CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

System Engineering and Informatics

Diploma Thesis

Web development of interactive map of Europe

Ada Arnautović

Declaration

I declare that I have worked on my master thesis titled "Web development of interactive map of Europe" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on 25. 11. 2016. ____

Acknowledgement

I would like to thank my supervisor Ing. Václav Lohr for his help and guidance during this project. I would also like to thank all my teachers and professors at every educational level, since the knowledge is accumulative. Special thanks to my mom for all the support she has always been giving me, especially during the time I have spent in Czech Republic. Finally, I would like to thank to my special friends who were there for me, whenever I would need them, offering friendly advices and providing the support.

Webové vývoj interaktivní mapě Evropy

Souhrn

Žijeme v éře neustálého zlepšování a nový vývoj v oblasti technologie. Není snadné sledovat všechny nové trendy a inovační koncepty a doplňující design styly. Vývoj webových stránek a design je významně ovlivněna nejnovější trendy a technologie. Vzhledem k tomu, změny a různé koncepty přicházejí neustále, uživatelé rovněž přijímají k nim rychlým tempem. Kromě toho vývojáři a návrháři mají přijmout ještě rychleji s cílem využít nové styly a nemělo by se nechat za hospodářskou soutěž. V návaznosti na některé pojmy a přístupy, může vést ke zvýšení on-line prodeje, pokud uživatelé přijímají je a cítí obsah o nich. Tento výzkum je veden k testování zda je či není interaktivity webových stránek, spolu se specifickým přístupem včetně "vzdělávací" a "Zábava" úseky by se zvýšil zájem uživatelů v on-line cestovní podnikání. Po vývoj a designu webových stránek cestování prototyp, cílové skupiny obyvatelstva je vyzván k vyplnění průzkumu a reagovat na otázky, které by pomohly učinit závěr. Odpovědi jsou analyzovány a grafický prezentovány a závěr byl na základě.

Po vývoji a konstrukci prototypu cestovní webové stránky, cílené populace je vyzván k vyplnění dotazníku a reagovat na otázeky, které by pomohly učinit závěr. Odpovědi jsou analyzovány a prezentovány graficky a závěr byl dělán založený na tom.

Klíčová slova:

web, vývoj, interaktivní média, kvíz, hra, turistika, cestování, Europe

Web development of interactive map of Europe

Summary

We are living in the era of the constant improvements and the new developments in the field of technology. It is not easy to follow up with all the new trends, innovative concepts and complementary design styles. Website development and design is significantly influenced by the newest trends and technologies available to be used. As the changes and different concepts are coming constantly, the users are also adopting to them in a fast pace. Moreover, developers and designers have to adopt even more rapidly in order to employ the new styles and not be left behind the competition. Following certain concepts and approaches can lead to increase in online sales, if the users accept them and feel content about them.

This research is conveyed to test whether or not interactivity of the website, along with the specific approach of including 'Educational' and 'Entertainment' sections would increase the interest of users in the online traveling business.

After development and design of the travelling website prototype, targeted population is asked to fill the survey and respond to set of questions that would help making a conclusion. The replies are analysed and graphically presented, and the conclusion was made based on that.

Keywords: web, development, interactive, media, quiz, game, tourism, travelling, Europe

Table of Contents

Table of Figures

1. Introduction

Nowadays, with all the technology we have around us, it would be impossible to imagine a world without it - Internet, web applications, computer games, news portals, and other available online features that became part of our daily routine and lives. Regular mail was being used instead of email, companies were promoted through certain brochures and catalogues rather than through their corporate websites, people were reading newspapers instead of news portals, social media was not an option as a way of communication, kids were playing regular games with each other, unlike today when there are so many devices and consoles available with millions of different games manufacturers offering even bigger variety of games themselves. Nonetheless, twenty-six years ago, a man named Tim Berners-Lee, changed the way people's lives were about to continue.

1.1 The origins of web design

Tim Berners-Lee is a British computer scientist and the inventor of the World Wide Web (WWW). He wrote the first web client and server in 1990 and he founded the World Wide Consortium (W3C), the forum for technical development of the Web, and the Web Foundation whose mission is that the WWW serves humanity. As a director of the W3C, a Web standards organization founded in 1994, Tim developed interoperable technologies, such as software, tools, specifications and guidelines to lead the Web to its full potential. He is also a director of the World Wide Web Foundation, which was launched in 2009 to coordinate efforts to further the potential of the Web to benefit humanity (1).

"The web was originally conceived and developed to meet the demand for automatic information-sharing between scientists in universities and institutes around the world. The first website at CERN - and in the world - was dedicated to the World Wide Web project itself and was hosted on Berners-Lee's NeXT computer. The website described the basic features of the web; how to access other people's documents and how to set up your own server" (2).

The website looked like nothing that exists today, there were no images, there was no Cascading Style Sheets (CSS), there was no parallax design; the website contained only the plain text. With the birth of Hyper Text Markup Language (HTML), a coded language that can be read by web browsers, and composed into web pages, the first website went live. Moreover, just a three years later W3C was established, aiming to implement a set of core principles and components so the Web can be accessed consistently by all (3). In order to understand the current trends and possible future developments and styles when it comes to the website design and the delivery of the content available online, it is important to know how did it all start and what were the changes that led us to what we have today.

1.2 History of web browsers

Up to 1993 all the websites were text heavy, because the early browsers did not support any images and only sixteen colors were supported by monitors and stated as a web safe colors. The first browser to support images was called Mosaic; it could display the text and the images together, and it was the first one to adopt the layout still favoured by today's browsers. Just a year later, another browser was created with the name Netscape Navigator, it supported on-the-fly page loading and it was so powerful that many websites carried the motif "Best if viewed in Netscape!" The same year, Opera 1.0 was launched as a research project. 1995 brings Internet Explorer (IE) 1 and 2, while 1996 is the year when Internet Explorer 3 and Netscape Navigator 3 were introduced. At one point, nearly 90% of the Internet users were navigating through Netscape Navigator, but since IE3 supported multimedia, Java applets, ActiveX controls and add-ons like Internet Mail it ate away at Netscape's market dominance to become the web's most popular browser for many years. During these years, design trends enhanced a bit from the very first browsers; number of web safe colors grew to 256, even though they were very bright and difficult to look at when comparing the ease with today's colors. In addition, web safe fonts were created, including simple serif fonts such as Times New Roman and Courier New. Moreover, first colourful buttons were created, which gave a 3D feel at that time. In addition, as the Internet grew in popularity, businesses and individuals looked at it as an advertising tool, trying to visually catch the reader's attention with garish, flashing advertisements. 1995-and 1996 were also the years when Personal Home Page (PHP), Cascading Style Sheets (CSS), Ruby, JavaScript, Active Server Pages (ASP), Multimedia Shockwave and software platform Flash were released. This created a variety of new options, including: making the creation of new, simple, dynamic web applications such as guest books and web forms possible, animation and interactivity on the web, exciting new layout and design possibilities, formation of the foundation for vector graphics, animation games and applications. With the introduction of HTML 4.0, in 1997, the web was internationalized and a Universal Character Set was defined to support a use worldwide. In the same year, the number of online Internet users grew to 100 million. From 1998 the content started being slowly scaled back, there was less text on the screen and more emphasis on menus and navigation; dark and black-colored backgrounds were popular. This year was famous for the Google launching. Over coming years, it redefined search and branches into countless other industries. Also, in 1998, CSS2 was released. 1999 brings Internet Explorer 5.0, while 2001 brings IE6. 2003 was the year when 3 technologies were lunched that are still in use every day, including: WordPress, an open-source blogging and content management system, MySpace the social platform, and Facebook, the social network. When it comes to design trends these years, there was an increasing emphasis on readability and functionality; Flash Animation Intro Pages were popular as animated content was subtly integrated into websites. Mozilla Firefox 1.0 was launched in 2004, as an open-source challenger to the ubiquitous Internet Explorer whish was still extremely popular. In 2005, the most famous video streaming site, YouTube was released, to be bought by Google just one year later. Long scrolling design, stock photography and dropping shadows were design trends for this period. 2007 was an important year when it comes to mobile technologies. It is the year when first iPhone was launched, redefining our relationships with our phones. Responsive design became more relevant. The same year Mobile Safari was released as a first browser for mobile phones. One year later, 2008, the first working draft of HTML5 was there as well as Google Chrome, which became an instant success story; now, it is the most widely used browser on the planet because of its speed and minimalistic design. Since 2010, the browser market was pretty much cornered around major players – Google Chrome, Mozilla Firefox, Internet Explorer, Safari and Opera. Between 2010 and 2014, Google released over 30 different Chrome updates, Firefox over 20, while IE, Opera and Safari release less frequent updates. The design trends changed to flat design, responsive design, parallax design, JavaScript animations and animated GIFs. 2011 brings 1 billionth website and there is unprecedented freedom of design (3). List of all the most used browsers through the history, with the years of release can be seen in the Figure 1.

Year	Browser
1990	Nexus
1993	Mosaic
1994	Netscape Navigator
1994	Opera 1.0
1995	Internet Explorer 1 and 2
1996	Internet Explorer 3
1996	Netscape Navigator 3
1999	Internet Explorer 5.0
2001	Internet Explorer 6.0
2004	Mozilla Firefox 1.0
2007	Mobile Safari
2008	Google Chrome

Figure 1 – Web browsers through history

1.3 Resolution and colors through history

Up to 1993 the common monitor resolution was 640x480. Moreover, only sixteen colors were supported by monitors and stated as a web safe colors. 800x600 was the dominant monitor resolution until 2000 for the most users (56%) but 25% of users started using 1024 x 768 resolution. Between 2002 and 2006 millions of users upscale from 800x600 to 1024x768 or higher. 1024x768 became the dominant display resolution. 600x480 is phased out altogether in 2005. In 2003, for the first time, over half's the Internet users view websites using 4 or 32bit hardware, displaying 16,722,216 different colors. By January 2010, more than three-quarters of users started browsing on a resolution higher than 1024x768. Few years later 1152x864 became the most-used resolution, which is still the case. List of the change and increase in the monitor resolution through the years can be seen in the Figure 2.

Year	Resolution
1990	640x480
1994	800x600
2004	1024x768
2014	1152x864
2015	1280x800

Figure 2 - Common resolutions through years

2. Objectives and Methodology

2.1 Objectives

The main objective is to create and develop a dynamic website with interactive map of Europe. The three most important aspects and approaches of this assignment are: education (Geography - providing interesting information about European countries), entertainment (game - creation a quiz to "test" the knowledge gained in educational part) and tourism (business oriented, i.e. travel agency).

In order to achieve the goal and make a meaningful observation (whose results will be interpreted to form an objective conclusion), the "problem" should be defined and a survey should be conducted. A problem might be defined in the following way: Does interactive media attract more potential customers/users than non-interactive one? Is the content the one which matters or people tend to be more attracted by the way that content is presented? Non-interactive entertainment has its own (emotional) delivery knowledge, but is that enough?

This assignment includes information, knowledge and programs/languages from the three different fields, including: Computer Graphics, Web Design and Graphics Design.

2.2 Methodology

Both, qualitative and quantitative approaches will be used in the methodology part of the project. For the qualitative approach, a survey will be made that will be answered by the targeted population. As for the quantitative approach, the results of the survey will be collected and analyzed using statistical methods. Based on the statistical analysis, importance of interactivity in the created website will be measured, compared to the static, non-interactive website with the similar content. These results would be important to conclude would a website as an interactive map of Europe bring potential travel agencies more customers and travellers then the already existing websites for travelling purposes.

Techniques used for methodology, such as empirical evidence (a source of information acquired by observation/experimentation) are important to acquire new knowledge or correct and integrate the previous one. That will be done in the following steps: identifying and defining the problem; accumulating relevant data; formulating a tentative hypothesis; conducting experiments to test the hypothesis and interpreting the results objectively.

3. Literature review

Web pages that are made only with HTML and CSS can display simply the static content. Static websites do not respond to user actions, i.e. pressing a key or clicking a mouse. In order to make a website dynamic and interactive, usually jQuery and JavaScript are used. "JavaScript is a programming language used to create web pages that change in response to user actions. jQuery is a collection of prewritten JavaScript code that lets us easily create interactive effects on our site" (4). Nonetheless, there is no website that can be made without HTML, and CSS (basically it can be done only with HTML, without CSS, but let's face it, that wouldn't be a case), so what follows is explanation of the technologies needed for creation of the interactive website.

3.1 HTML

Hyper Text Markup Language, or shortly, HTML is, as its name suggests, the markup language, which allows developers to construct visible or audible web pages that are rendered by web browsers. The word Hyper Text denotes a method for the linkage of the different, separated, web pages, into one website. Basically, every website that we can possibly access is written in HTML, so it can be described as a skeleton that gives the structure to the websites. It is nothing more than a text documents with different tags, which when being marked up, tells the browser how to display the web page.

HTML is structured in the way that all the elements are written inside angle brackets, such as <p>. These tags have to be properly closed, once there are open, so they are usually used in pairs, like <title> and </title>. Nevertheless, there are few exceptions where the opened tag does not need the closing match, like is the case with <img>. Web browsers do not display HTML tags; instead, they use them to translate the exact content of the page. (5)

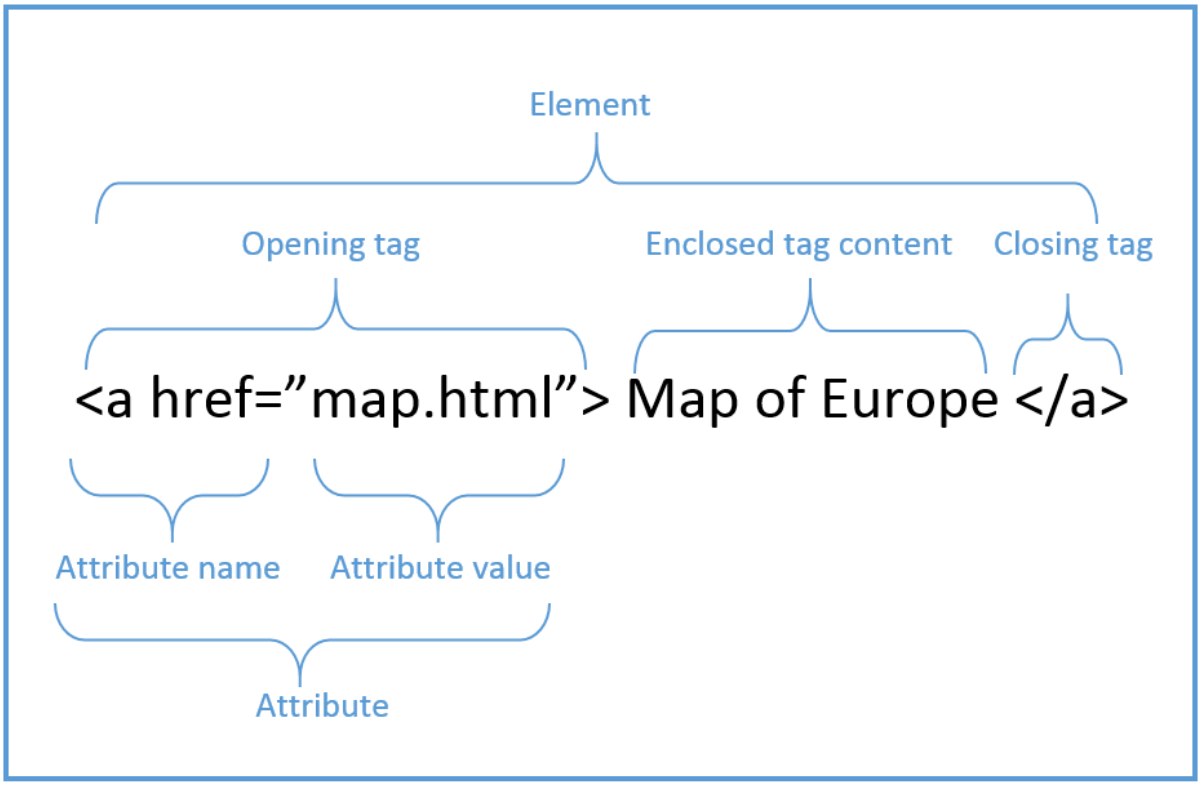


Figure 3 – Breakdown of an HTML tag

As it can be seen in a Figure 3, an HTML tag consists of an opening tag, enclosed tag content and a closing tag. An attribute usually lies between the opening tag, including attribute name and attribute value. Both, opening and closing tags, together with the content among them are considered as a one whole element.

HTML was originally created by Tim Berners-Lee in 1991, with the intent to define the structure of documents such as lists, paragraphs and similar, in order to facilitate the sharing of precise information between scientists. However, the first standard specification was released in 1995, called HTML 2.0 and it already started being widely used to shape websites, using the available elements (6). Four years later, a major version comes, HTML 4.01. This was a widely used version, for more than 10 years, just until HTML 5 was published, as the extension for the HTML 4.01, in 2012.

3.1.1 HTML 5

It can be said that before HTML5, the earlier versions, were not able to provide much of the interactivity in the web domain. Fortunately, that was about to change with the 5th version; that is still nothing more than the markup language, used for structuring and presenting the web pages. Nevertheless, the main goal of HTML5 is to enhance the language by providing support for the most recent multimedia, and at the same time still keeping it user friendly for people and reliably understood by different machines and devices. It is important to mention that all the modern browsers listed in the Figure 1 started supporting HTML5, while HTML works smoothly with all old browsers. The main new features of HTML5, when it comes to multimedia and interactivity, which did not exist in the previous specifications are: audio and video, vector graphics (SVG and canvas), JavaScript (being allowed to run in the background -Web worker API) (5). Since new features that are available in HTML5 help to increase interactivity on the web, they will be explained separately:

- <canvas> – a resolution-dependent bitmap canvas that provides an easy way to draw and render game graphics, graphs or other visual images using JavaScript, without the need for the Flash code. Moreover, it enables the creation of animations of various complexity. "A canvas is a rectangle in your page where you can use JavaScript to draw anything you want" (7). HTML5 specification contain the canvas API, which is a set of functions used for drawing shapes, defining paths, making gradients and applying transformations.
- <svg> - Scalable Vector Graphics is a vector-based image format, used to define 2D Web graphics. Since it is a vector, it provides a possibility of image scalability without reducing the image quality. Behaviour of these images is defined in XML files, which means that they can be accessed through a text file. Advantages of SVG over JPG, GIF, PNG and similar, include possibilities for: creation and modification using JavaScript, printing with high quality regardless of the resolution, animation using the built-in elements, containing hyperlinks to the remaining documents, etc. In comparison to canvas, the main difference would be that SVG images are vector-based, meaning that they are composed of a fixed set of shapes defined by math, while canvas manipulates pixels (8).

- <video> - a tag that made possible to embed a video to the website without usage of the third-party plugins. Most likely the most famous third-party video plugin was Adobe Flash, which is not an open source plugin, so it makes a huge difference playing a video only through browser. Even though this is possible with the supporting browser, player controls, video dimensions and a poster image only, it is more complicated than it sounds. Complications were caused because of the different formats and fragmentation. Google Chrome, Mozilla Firefox, Opera, Safari and Internet Explorer as major players in the game were not unified in the decision to support either WebM, which is a royalty-free format, or ubiquitous but costly MP4 format. Nonetheless, these days there is support for HTML5 video and the related formats (9). Figure 5 shows the list of video formats with the browsers that support them.

Video	**Ogg**	**MP4**	**WebM**
IE	No	Yes	No
Chrome	Yes	Yes	Yes
Firefox	Yes	Yes	Yes
Safari	No	Yes	No
Opera	Yes	Yes	Yes

Figure 4 – HTML5 Audio formats supported by browsers

- <audio> - a great and easy-to-implement tool that enables web developers to embed audio in websites unlike before HTML5. The audio tag is very simple, consisting of: obligatory source tag which is used to link to the actual audio file; optional controls attribute which is used to communicate with browser, in order to create audio controls and place them exactly where the audio tags are. If this attribute would not exist, it would be impossible for the user to play and/or pause the sound. It is a good practice to use two source tags, for cross-browser support purposes, since not the all browsers do support all the audio files (10). The list of the browsers and the audio formats they do support is shown in Figure 4.

Audio	MP3	Wav	Ogg
IE	Yes	No	No
Chrome	Yes	Yes	Yes
Firefox	Yes	Yes	Yes
Safari	Yes	Yes	No
Opera	Yes	No	Yes

Figure 5 – HTML5 Video formats supported by browsers

- JavaScript – unlike HTML4 and earlier versions when JavaScript was not allowed to run in the browser, but in same thread as browser interface, in HTML5 JavaSript can run in background, thanks to the JavaScript Web Worker API. Similarly, before HTML5 it was barely possible to get true GeoLocation of the user browsing the website, and this would be even bigger issue when it comes to the mobile phones. Luckily, this feature is improved a lot with a help of JavaScript GeoLocation API, that is available in HTML5 and it does help identify the user's location while browsing any website, given that, of course, the user does allow it (11).

3.1.2 XML

Extensible Markup Language (XML) is used for documents which contain structured information. Structured information means, that there should be some kind of content, i.e. text, image, etc. and a suggestion or clue that indicates the content's role – even though if the content is of the same type, let's say text, it would have a different role when placed in a database, or as a figure caption, or in a heading of the section. "A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents" (12).

The very often misunderstanding is that XML is no different from HTML, which is not true. Tag semantics is fixed in HTML, unlike XML; <h1> will always be a first level heading, and <h6> will always be the last level heading, while, for example, <book.title> would be meaningless. "The W3C, in conjunction with browser vendors and the WWW community, is constantly working to extend the definition of HTML to allow new tags to keep pace with changing technology and to bring variations in presentation (stylesheets) to the Web. However, these changes are always rigidly confined by what the browser vendors have implemented and by the fact that backward compatibility is paramount" (12). On the other hand, XML does not have specified semantics neither tag sets, which confirms once again that it is a real meta language, meant to describe the markup languages. This means that tags, together with relationship between them, are not predefined; moreover, the whole semantics is defined by the applications that process an XML document, or by the stylesheets (12).

3.1.3 XHTML

Extensible HyperText Markup Language (XHTML) is a mixture of HTML4 and XML. Basically, it is till the markup language based on HTML4, but with power of XML added to the website. XHTML can be also called a strict HTML because unlike in earlier versions there are some rules that have to be respected, such as:

- a doctype indicator must be included in all documents
- tag names must be in lowercase

- all documents must be properly structured
- all tags must be properly nested
- all tags must be closed
- all attributes must be quoted
- the <title> tag must come first in the <head> tag.

Because of these stricter rules, the code is cleaner and more self-explanatory. In addition, XHTML provides more compatibility and conformity. To be more precise, this means that the code is clean and formatted, which makes the website load faster. With greater control over documents displayed in various browsers and different devices, which is provided by XHTML, there is a better accessibility, which eventually should lead to more page views.

3.2 CSS

Content-wise, as HTML is used for the meaning, CSS is used for the presentation. First, the content is structured using HTML, then it is visually formatted using CSS.

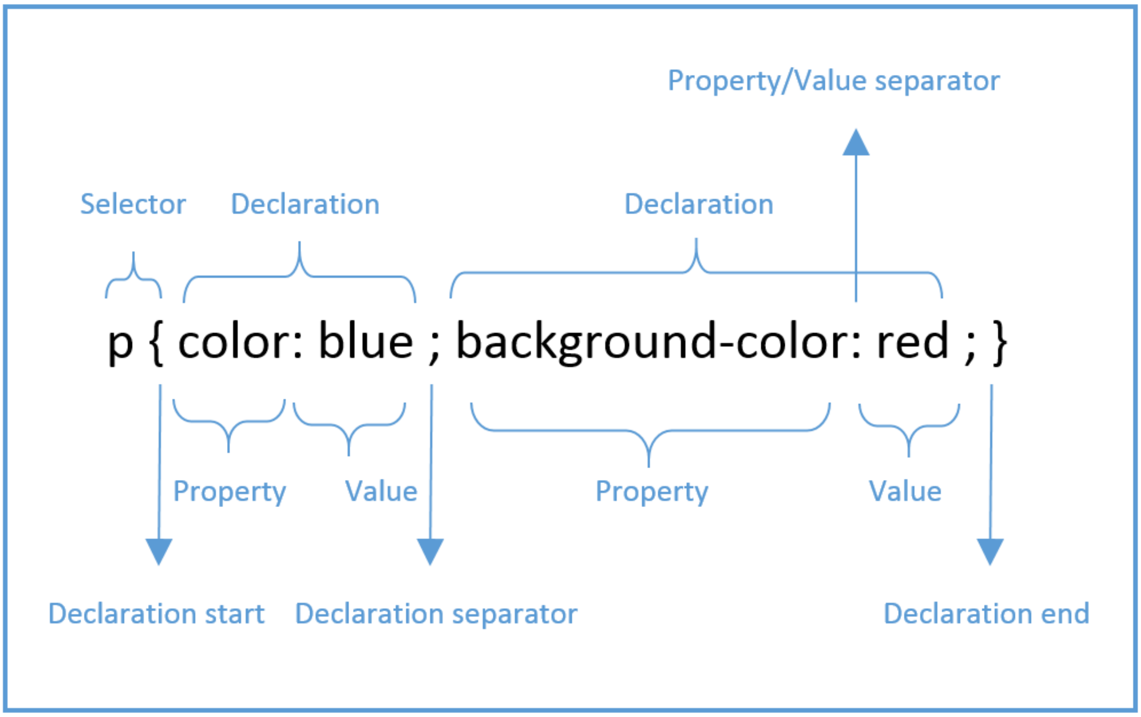


Figure 6 – CSS rule set

Styles are not similar in breakdown to HTML, they have a format of ‘property: value’ and most of the properties can be applied to the majority of the HTML tags. Figure 6 shows the CSS rule set. It consists of a selector and declarations, followed by a declaration start and declaration end and eventually a declaration separator in a case of more declarations. Each declaration contains a property, value and a property/value separator. There are three possible types of selectors:

- tag – like in the Figure 6, an HTML tag such as p
- class - a class attribute of one or more elements, such as
  <p class="class-name">This is a paragraph</p>
- ID - An id attribute of a unique element, should only be used once, such as <div id="header">This is a header<div> (14).

In order for a CSS to affect the HTML document, they must be linked together. There are three ways to combine the two:

- Inline style – the style attributes containing CSS properties should be added to the relevant document; it might be used to apply a unique style for a single element
- Internal style sheet – styles should be defined within the <style> tag, in the <head> section of an HTML page; it may be used if a single page has a unique style
- External style sheet – the style sheet file should not contain any HTML tags and it has to be saved as a .css document; the look of an entire website might be changed by updating one single file (15)

3.2.1 CSS3

The good news is that CSS3 is completely backwards compatible with CSS2, while at the same time it offers new features. The main advantages that CSS3 brings are modules. Modules are different sections that CSS3 is split up into that go through W3C in various stages of the recommendation process. Unlike earlier versions, CSS3 does not submit all the content as a single document, but it breaks it into smaller, individual pieces – modules. Thanks to this, there is a much wider range of browser support. Apart from modules, there are new CSS selectors, some new pseudo-elements and a number of new CSS properties. The purpose of most of the new properties is to create visual styles that can likely associate more with graphics programs, like some from Adobe Creative Suite (i.e. Adobe Photoshop). Border-radius and box-shadow are examples of such properties (16).

3.3 JavaScript

JavaScript is a dynamic programming language, designed for creating network-centric applications. It is integrated with Java and HTML, which makes it easy to implement. This open and cross-platform computer language enables addition of powerful interactions to the web pages. JavaScript appeared for the first time at 1995, in Netscape 2.0 with the different name: LiveScript (17). It can be said that almost everything in JavaScript acts like an object, which is unordered collection of properties, each of which has a name and value. These objects are dynamic, which means that their properties can usually be added and deleted, and they are created by invoking a constructor function with the new keyword, or by using a shorthand literal expression. In addition, constructor functions are objects; therefore, objects create objects (18).

With the addition of the client-side JavaScript, static HTML pages might become dynamic and interactive. Once the script is linked with the HTML document, the browser starts interpreting the code. Client-side JavaScript offers many benefits over traditional, server-side scripts. For instance, it may be used to check the validity of an e-mail address entered by the user. After the user submits the web form, only if all the fields were valid, they would be forwarded to the server.

Advantages of using JavaScript might be summarized as following:

- Less server interaction – user's input can be validated before sending it to the server, which would save some of the server's traffic
- Immediate feedback to the visitors – users do not need to wait for the website to reload to check if they have not entered something that they had to
- Increased interactivity – interactive interfaces can be created; i.e. there can be a certain reaction when a user activates them via keyboard or the mouse

- Richer interfaces – with the usage of JavaScript, various components such as sliders and drag-and-drop can be included to offer a rich interface to the users (17).

JavaScript is the foundation for many commonly used libraries and frameworks, such as jQuery, Angular.js, Node.js, Reacct.js and similar.

3.3.1 JavaScript Libraries and Frameworks

JavaScript library is a collection of pre-written JavaScript code that allows simpler development of web applications. It can also be described as a toolkit that serves to abstract different layers, such as web browsers, Document Object Models and similar. A purpose of a good JavaScript library is to offer a variety of different tools to work with, which should simplify the complete coding process. One of the most used JavaScript libraries is jQuery.

On the other hand, JavaScript framework serves to provide a certain structure for the way the code should be presented. Framework or sort of a code template solves and/or simplifies the certain problem, while bringing the architecture in the order as well. Some of the most famous JavaScript frameworks are backbone.js, node.js, angular.js, react.js etc.

3.3.1.1 jQuery

"jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript" (19).

The newest version available at the moment is jQuery v.3.1.1. It has lightweight footprint, with only 32kB; it is CSS3 compliant – it supports CSS3 selectors to find elements as well as in style property manipulation and it has cross-browser support - Chrome, Edge, Firefox, Safari, Android, iOs and more (19).

3.3.1 Events

The main responsible feature that enables dynamic content and interaction with users is JavaScript event. In order to make the interaction possible, there has to be a way of detecting user actions, so the reaction can be prepared at time. Moreover, specific functions to be executed have to be known; the ones that would increase the appeal of the website. In the most cases, the events occur when the user 'makes a move'. Exceptions are the events that are not directly triggered by the user, such as load event, for example. Nonetheless, thanks to the event handler, some of those events can be detected by JavaScript itself. From the early browser days, starting from Netscape Navigator 2, event handler could have been attached to the specific HTML elements. It would wait until an event takes place and after happening, it would handle it by executing pre-defined JavaScript script. "When the user takes action he causes an event. When your script makes the page react to this event, interactivity is born" (22).

Event types		
Interface events	Mouse events	Form events
load	click	submit
unload	dblclick	reset
resize	mousedown	
scroll	mouseup	
focus	mousemove	
blur	mouseover	

	mouseout	

Figure 7 – JavaScript events

3.4 Bootstrap

Bootstrap is a front-end framework that aims to aid in website design. This free and open-source technology consists of a many templates based on HTML and CSS for visual design. Also, many optional JavaScript extensions are included. “Originally created by a designer and a developer at Twitter, Bootstrap has become one of the most popular front-end frameworks and open source projects in the world” (20). It was originally developed as a Twitter framework with main focus to assure consistency. Through the six years of Bootstrap’s existence, various versions were being released. The last one was Bootstrap 4, being publically available from August, 2015. One of the most important features of this framework, is that the latest version of it is always supported by the latest versions of all the major browsers used throughout the world. Another key features, there were employed in the second and third versions of Bootstrap are “responsive web design” and “mobile-first design” philosophies. This means that the layout of the web page, together with all the content will adjust logically by default, according to the screen size of the device that the website is being open on. This open-source framework encourages all the developers to participate and contribute in the further development.

3.5 DOM Manipulation

Document Object Model (DOM) is a programming interface that treats valid HTML, XHTML and well-formed XML documents as a tree structure, where each node is an object representing a part of the document. “It defines the logical structure of documents and the way a document is accessed and manipulated” (21). The browser creates a DOM once the website is loaded.

The HTML DOM model is constructed as a tree of objects, which can be seen in the figure below.

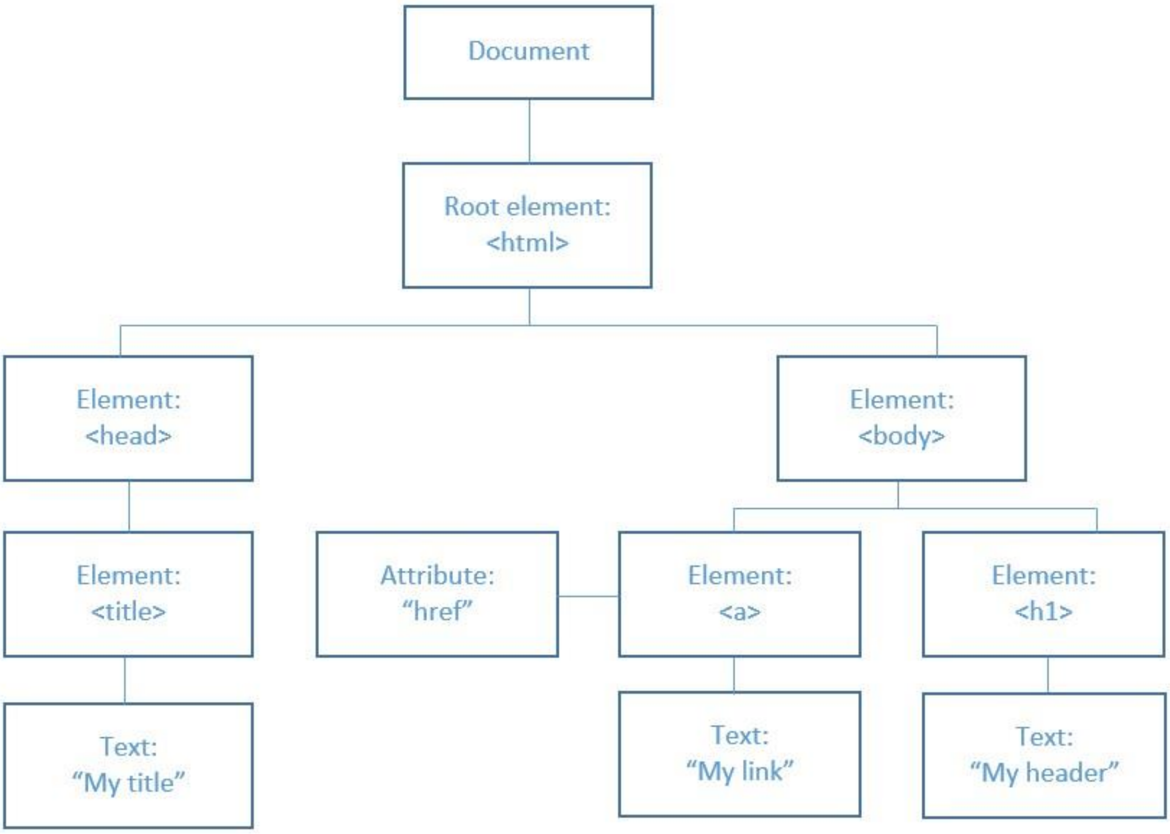


Figure 8 – HTML DOM model

Figure 8 is a typical HTML tree structure. The first and main node is the document node, which extends to a node with root element - <html>. The root element contains two main parts of the document, <head> and <body>. Further on, <head> and <body> can be extended to a set of various elements, such as link, heading title etc. that can be connected with a specific attribute, i.e. "href" (22).

HTML DOM can be accessed with any programming language, including JavaScript. All the elements are considered objects, whose interface includes properties – values that can be set or changed; and methods – actions that can be done (22).

3.6 Interactive Media

If we go back when everything started, we can find a quote from man that is responsible for creation of what we know as World Wide Web today:

“The Web as I envisaged it, we have not seen it yet. The future is still so much bigger than the past. You affect the world by what you browse.”
Tim-Berners Lee

Starting with this, it can be clearly understood that it is not easy to predict the future standards, features and eventually design of the Web. For Web developers, Web designers and all the other people involved in the process of the website’s creation, the learning process never seems to end. This is mainly because the new technologies are developing at a very fast pace, while at the same time multimedia manipulation seems to be boundless. Programming languages continuously tend to develop and grow more and at the same time design standards are being upgraded in an effort to keep it up with everything else.
“For Web developers, the learning process never seems to end. New technologies are constantly being developed, the manipulation of multimedia seems to know no bounds, the proliferation of knowledge, distribution and programming languages continue to develop and grow, and, of course, design standards are constantly being upgraded in an effort to keep up with it all” (24).
Interactive media can be defined as an incorporation of digital media, such as graphics, sound, images and text into a digital computerised environment – mainly Internet, that enables data interaction. Multimedia, new media, interactive media, as well as interactive design are all the names that can be found thinking of the same thing [25].

4 Practical Part

Practical part consists of a website that is created and functioning using XAMPP server, HTML5, CSS3, JavaScript, along with some of the libraries and frameworks such as jQuery, Bootstrap and Raphael. XAMPP is an open source, cross-platform web server package that stands for Cross platform/Apache HTTP server/MySQL/PHP/Perl. Its main purpose is to help developers and designers to evaluate their code on their computers without the need to connect to the Web.

As described in literature review, Hyper Text Markup Language (HTML) allows the construction of the visible and/or audible websites that can be rendered by the web browsers. While HTML5 is the newest version of HTML and it supports Scalable Vector Graphics (SVG) that are used in the creation of this website. While HTML is used for the content, on the other side Cascading Style Sheet (CSS) is used for the presentation and the visual formatting. CSS3 is the latest version of CSS that offers some new features, that have been used in the creation of this website, such as changing the opacity of the certain element and adding the shadows around it. Last but not least, JavaScript and its libraries were used to make interactivity and to enrich this static HTML and CSS content with dynamic features.

The website has five main tabs, that a user can navigate through, including: Home – place where the map is, Geography – list of the countries with links to their pages, Quiz – a game application where the users can have fun and test their knowledge, Travel – a sample web page for the travel agencies and Contact – information about this thesis.

4.1 Home

Home is the first tab and the homepage that opens by default when the website is visited. It has very minimalistic style, consisting of the navigation menu and the interactive map of Europe.

4.1.1 Navigation menu

The navigation menu consists of the five tiles, each corresponding to, Home, Geography, Quiz, Travel, Contact, retrospectively. The final look of the navigation menu can be seen in the Figure 9 below:

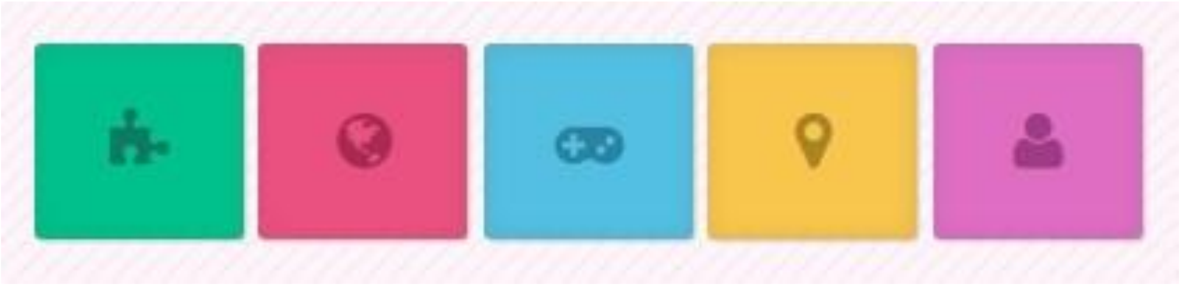


Figure 9 - The navigation menu

Creation of the navigation menu did not take more than 32 lines in the HTML code, as it can be seen in the Figure 11. The whole menu is wrapped in the <nav> tag, and place into one big unordered list - <ul>, which contains five different lists - <li> based on the color. Furthermore, each of the lists has their own unordered lists, that contain the possible choices of the dropdown menu. The idea behind this menu is that the tiles themselves are not clickable, meaning that they would redirect to the same page, which can be understood from the Figure 11 in the part <a href=”#” class=”…” ></a>, but their menus are clickable, directing to the desired destination. Figure 10 shows the look of the menu when hovering on the middle element, which has the gamepad icon. What happens is that a small dropdown tile appears, offering the clickable options that can bring the user to the another web page. In this case, the only option is a tile with the “Quiz” content, and as seen from the Figure 11, the code responsible for that redirection is:

```
 <li class=”blue”>
     <a href=”#” class=”icon-gamepad”></a>
           <ul>
                      <li>
                              <a href=” navs/quiz.html”>Quiz </a>
                       </li>
           </ul>
</li>
```

Explanation of this code is as follows: everything is wrapped in the list with has style: blue; reason for that is easier visual formatting in CSS that is shown and explained in the Figure 12. Blue tile does not have any link, its anchor is empty, but it has class “icon-gamepad”, which gives this small icon of the joystick, via Font Awesome, that is included in the <head> part of the HTML document. Below the tile there is <ul> that can contain as many fields as wanted, but in this case it is only one. Furthermore, the one existing element in the list has one element that is named “Quiz” which is the text we can read from the menu item, and when clicks it directs to the document called quiz.html, placed in the navs folder.

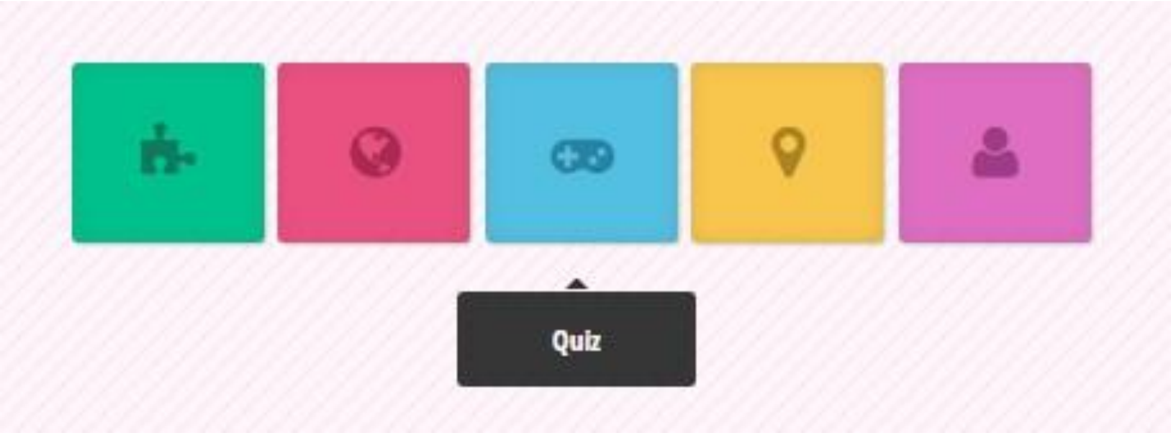
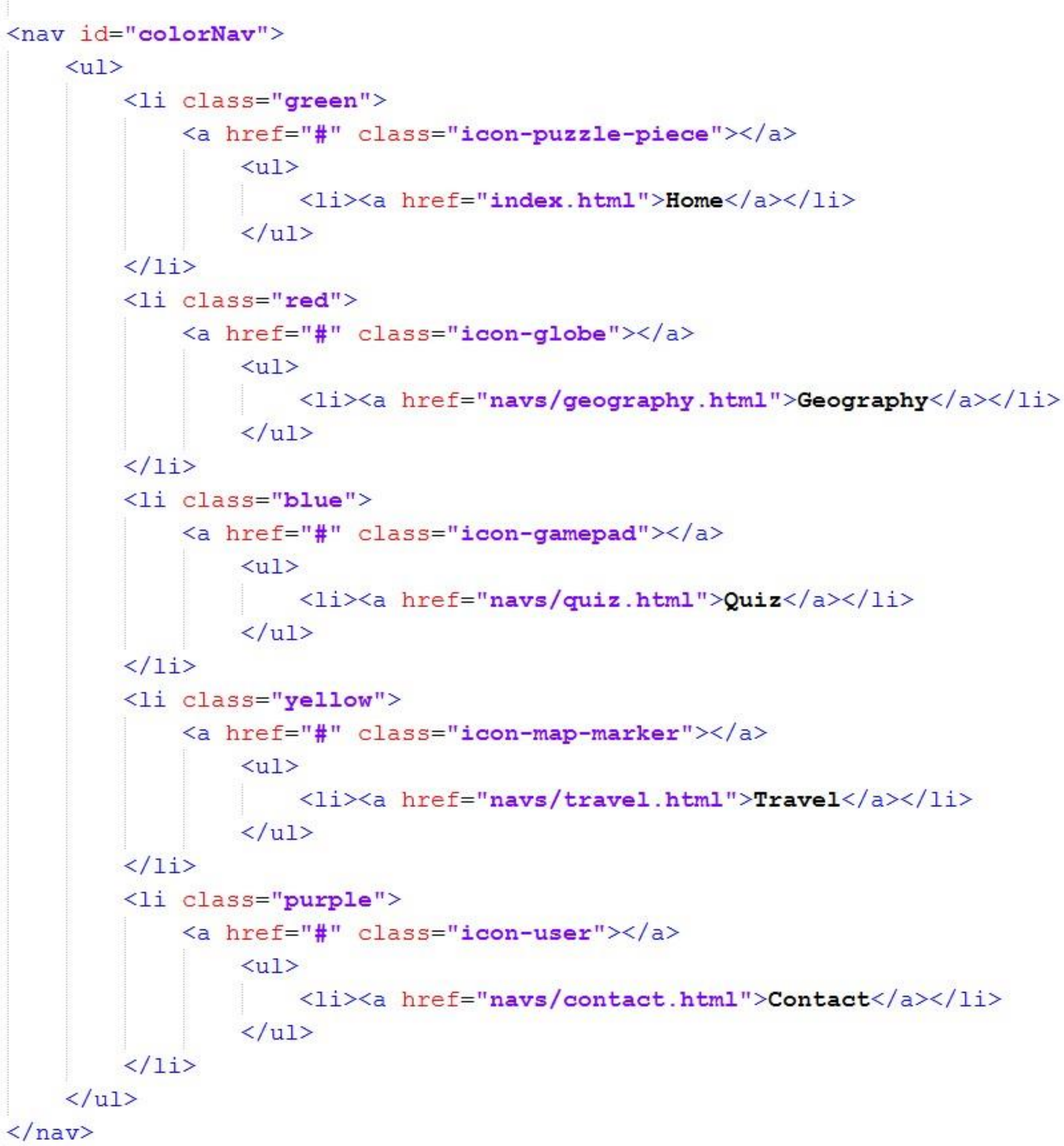


Figure 10 - Navigation menu dropdown

With similar logic, the rest of the tiles for the navigation menu were created. It happened to be a case that there was no need for more elements dropping below the main tile, which is the reason why every unordered list with icon classes (five main tiles from the menu) has only one element in the list below it. Also, different classes that can be seen in the code, such as “icon-puzzle-piece”, “icon-globe”, “icon-gamepad”, “icon-map-marker” and “icon-user” are available thanks to the Font Awesome, that is a collection of 675 scalable vector icons that can be instantly customized – size, color, drop shadow and other features, only with the power of CSS, without JavaScript required.



```
<nav id="colorNav">
    <ul>
        <li class="green">
            <a href="#" class="icon-puzzle-piece"></a>
                <ul>
                    <li><a href="index.html">Home</a></li>
                </ul>
        </li>
        <li class="red">
            <a href="#" class="icon-globe"></a>
                <ul>
                    <li><a href="navs/geography.html">Geography</a></li>
                </ul>
        </li>
        <li class="blue">
            <a href="#" class="icon-gamepad"></a>
                <ul>
                    <li><a href="navs/quiz.html">Quiz</a></li>
                </ul>
        </li>
        <li class="yellow">
            <a href="#" class="icon-map-marker"></a>
                <ul>
                    <li><a href="navs/travel.html">Travel</a></li>
                </ul>
        </li>
        <li class="purple">
            <a href="#" class="icon-user"></a>
                <ul>
                    <li><a href="navs/contact.html">Contact</a></li>
                </ul>
        </li>
    </ul>
</nav>
```

Figure 11 - HTML code for the navigation menu

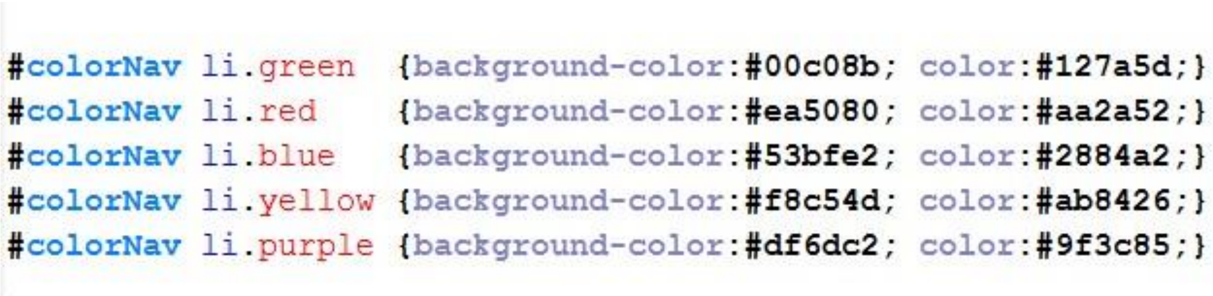
One of the ways to include font awesome, and to be able to use all the icons they are offering, is to download it from http://fontawesome.io/ and to link a document as a .css file to the original .html file.

Since whole HTML code for the creation of the navigation menu is clear and explained, the only remaining part is the CSS code. The part of it defining colors is shown in the Figure 12. Since there are five main tiles in the menu itself, there are also five classes for the lists defined in CSS. Names of the classes are green, red, blue, yellow and purple – based on the desired color of the tile, and properties being defined are background color and color itself. Background color is the lighter hue of the green/red/blue/yellow/purple that is used for the tile, while the color is the darker hue, that is used for the icon. For example, the first tile, representing the Homepage, with the puzzle icon, has light green background (in hex: #00c08b), and dark green icon (hex: #127a5d), written as:

#colorNav li.green {background-color:#00c08b; color: #127a5d;}

These five colors are also used and defined as the primary colors, along with the black, white and shades of grey throughout the whole document, to keep the color consistency.

There is more of CSS related to the navigation menu than these 5 lines of codes regarding colors, and it defines various properties, such as position, width, margin, padding, font size, text decoration, position, list style, text align, max height, overflow, display, line height, border, border radius and many others, in order to make it responsive and displaying elegantly on any device size, and all of those can be found in Appendix 1.



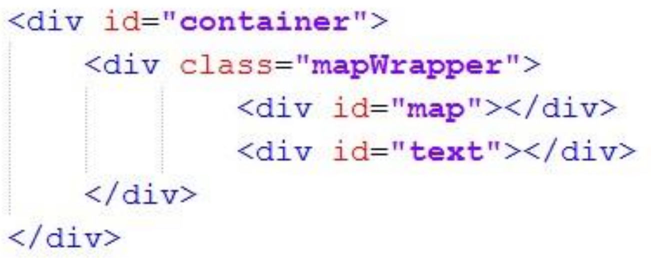
```
#colorNav li.green  {background-color:#00c08b; color:#127a5d;}
#colorNav li.red    {background-color:#ea5080; color:#aa2a52;}
#colorNav li.blue   {background-color:#53bfe2; color:#2884a2;}
#colorNav li.yellow {background-color:#f8c54d; color:#ab8426;}
#colorNav li.purple {background-color:#df6dc2; color:#9f3c85;}
```

Figure 12 - CSS code for the navigation menu

4.1.2 Map of Europe

The map contains all the European countries, including: Albania, Andorra, Austria, Belarus, Belgium, Bosnia and Herzegovina, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Great Britain, Hungary, Ireland, Island, Italy, Kosovo, Latvia, Lichtenstein, Lithuania, Luxembourg, Macedonia, Malta, Moldova, Montenegro, Netherlands, Norway, Poland, Portugal, Romania, Russia, San Marino, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey, Ukraine and Vatican. Some northern African – Morocco, Algeria, Tunisia, Libya, Egypt and western Asian countries – Israel, Lebanon, Jordan, Syria can also be seen on the map, because of their geographical position, but they are set to be off since they are not the topic of this research.

Figure 16 shows the look of the map by default, without any country being selected or hovered on. In order for it to appear as it does, various documents were created, including: HTML code, which consists only of 6 lines of code, as it can be seen in the Figure 13.

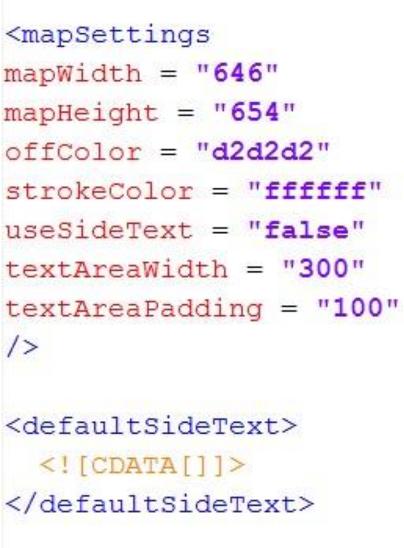


```
<div id="container">
    <div class="mapWrapper">
            <div id="map"></div>
            <div id="text"></div>
    </div>
</div>
```

Figure 13 - HTML code for map of Europe

Apart from HTML code, there is as well CSS code, that defines CSS properties of class mapWrapper and of map and text <div>. Container <div> is part of the Bootstrap specifications, and it is used as so, to remain scalable and responsive. The whole CSS code for the map can be found in the Appendix 1. Moreover, there is an XML document, that contains all the necessary information about every country on the map, including the name tag that will be shown once the mouse hovers over the country, state mode that can be ON or OFF which is in this case ON for every European country and OFF for all the countries located around, initial state color (the color of each country as it is, before being hovered or clicked on – that is set to be dark grey by default for all the counties), state over color (the color that changes from initial dark grey to the one of five primary colors based on the country location) and selected color (the color after the specific country has been clicked on).

Figure 14 shows the general XML code consists of 12 lines of code and applies for the look of the whole map. Except for the main settings, such as width, height, colors and paddings, there is also an option to use the side text, which would mean that once the specific country is selected, the information about it would be presented on the side. In this project that option is set to be false, because countries are presented in a separate page.

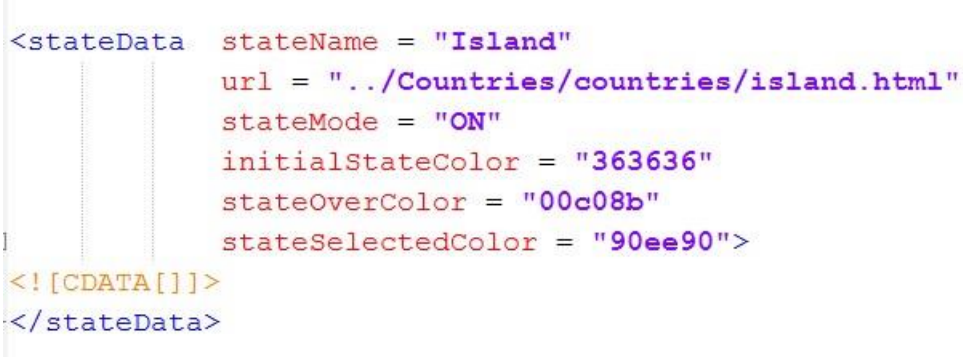


```
<mapSettings
mapWidth = "646"
mapHeight = "654"
offColor = "d2d2d2"
strokeColor = "ffffff"
useSideText = "false"
textAreaWidth = "300"
textAreaPadding = "100"
/>

<defaultSideText>
  <![CDATA[]]>
</defaultSideText>
```

Figure 14 - XML code for the map – general

Figure 15 shows the XML code for the specific country, in this example Island. Apart from that, the full XML code for all the countries can be found in Appendix 2.



```
<stateData   stateName = "Island"
             url = "../Countries/countries/island.html"
             stateMode = "ON"
             initialStateColor = "363636"
             stateOverColor = "00c08b"
             stateSelectedColor = "90ee90">
<![CDATA[]]>
</stateData>
```

Figure 15 - XML code for the map - specific country

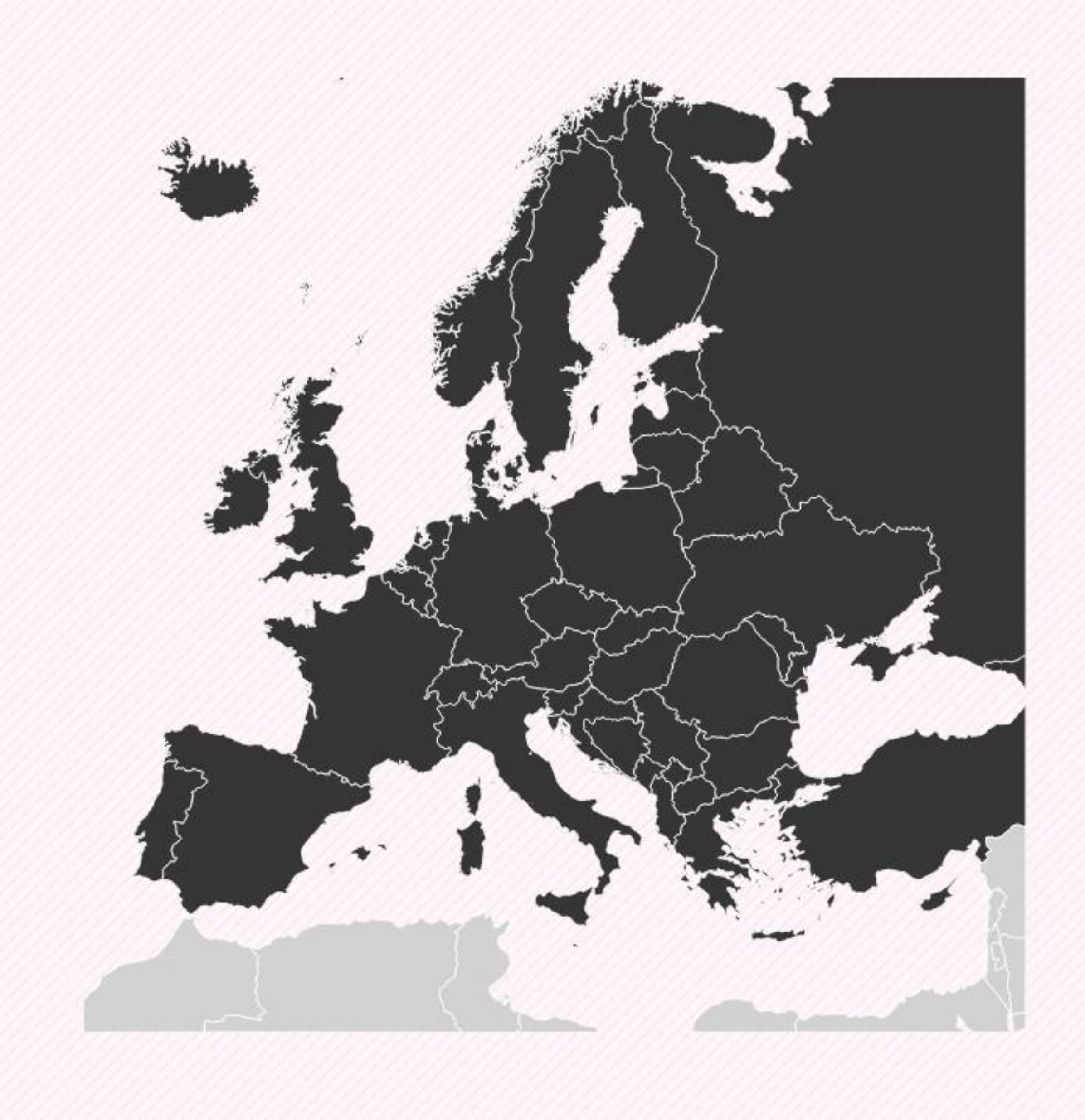
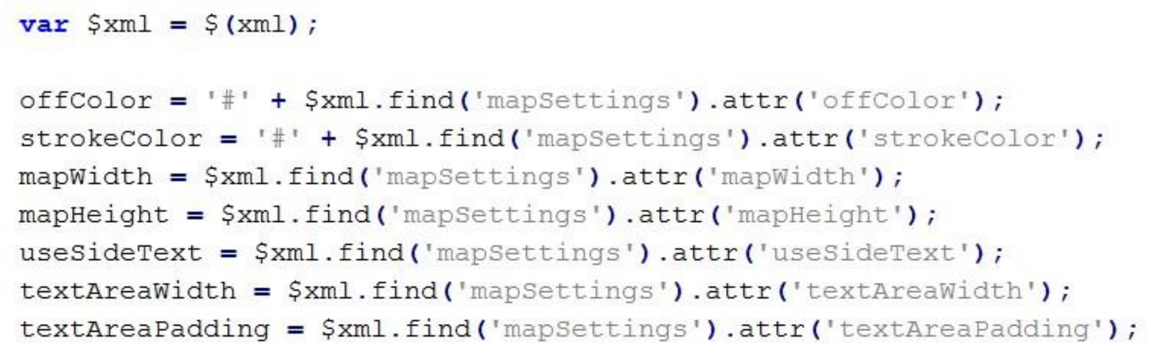


Figure 16 - Map of Europe; default look

It is important to mention that for the creation of this map, apart from HTML, CSS and XML, the JavaScript was used as well. Two famous JavaScript frameworks, jQuery and Rapheal, together with the another JavaScript document that was written to manipulate the coordinates were integrated in order for this map to look and work the way it does. Figure 17 shows the sample of the JavaScript code written in order to manipulate already existing XML code, and to integrate it with HTML and CSS in the desired way. As it can be seen, properties such as offColor, strokeColor, mapWidth, mapHeight, useSideText, textAreaWidth and textAreaPadding were being read from XML and imported to the JavaScript in order for the desired outcome to be visible. The whole code written for the map manipulation can be found in the Appendix 3.



```
var $xml = $(xml);

offColor = '#' + $xml.find('mapSettings').attr('offColor');
strokeColor = '#' + $xml.find('mapSettings').attr('strokeColor');
mapWidth = $xml.find('mapSettings').attr('mapWidth');
mapHeight = $xml.find('mapSettings').attr('mapHeight');
useSideText = $xml.find('mapSettings').attr('useSideText');
textAreaWidth = $xml.find('mapSettings').attr('textAreaWidth');
textAreaPadding = $xml.find('mapSettings').attr('textAreaPadding');
```

Figure 17 - JavaScript code manipulating XML

Figure 18 shows the look of the map when specific country is being hovered on by the mouse. In this case, it was Czech Republic, it color changed to light red, and the name tag popped out above the country. According to the primary colors scheme, the countries were divided into five categories, based on their geographic location: western, eastern, southern, northern and central. Logically, Czech Republic found its place in the category of the central European countries, along with: Slovakia, Poland, Hungary, Austria and Slovenia. All of these countries belong to the “red” primary color, hence when hovered over them, this pinkish hue of red will take over the initial dark grey. The southern countries belong into “purple” category, including the south Balkan countries: Bosnia and Herzegovina, Croatia, Serbia, Montenegro, Macedonia, Kosovo and Albania, along their western neighbours: Italy, San Marino, Vatican, Spain, Portugal, Andorra, France and Malta and eastern southern European countries: Greece, Cyprus and Turkey. Northern countries fit into the “green” category, with total of six countries belonging there: Island, Norway, Finland, Sweden, Denmark and Estonia. “Blue” category is reserved for the western European countries, including: Great Britain, Ireland, Germany, Belgium, Netherlands and Switzerland. The last but not least is the “yellow” category, with eastern European countries, listing: Russia, Belarus, Ukraine, Bulgaria, Moldova, Romania, Latvia and Lithuania.

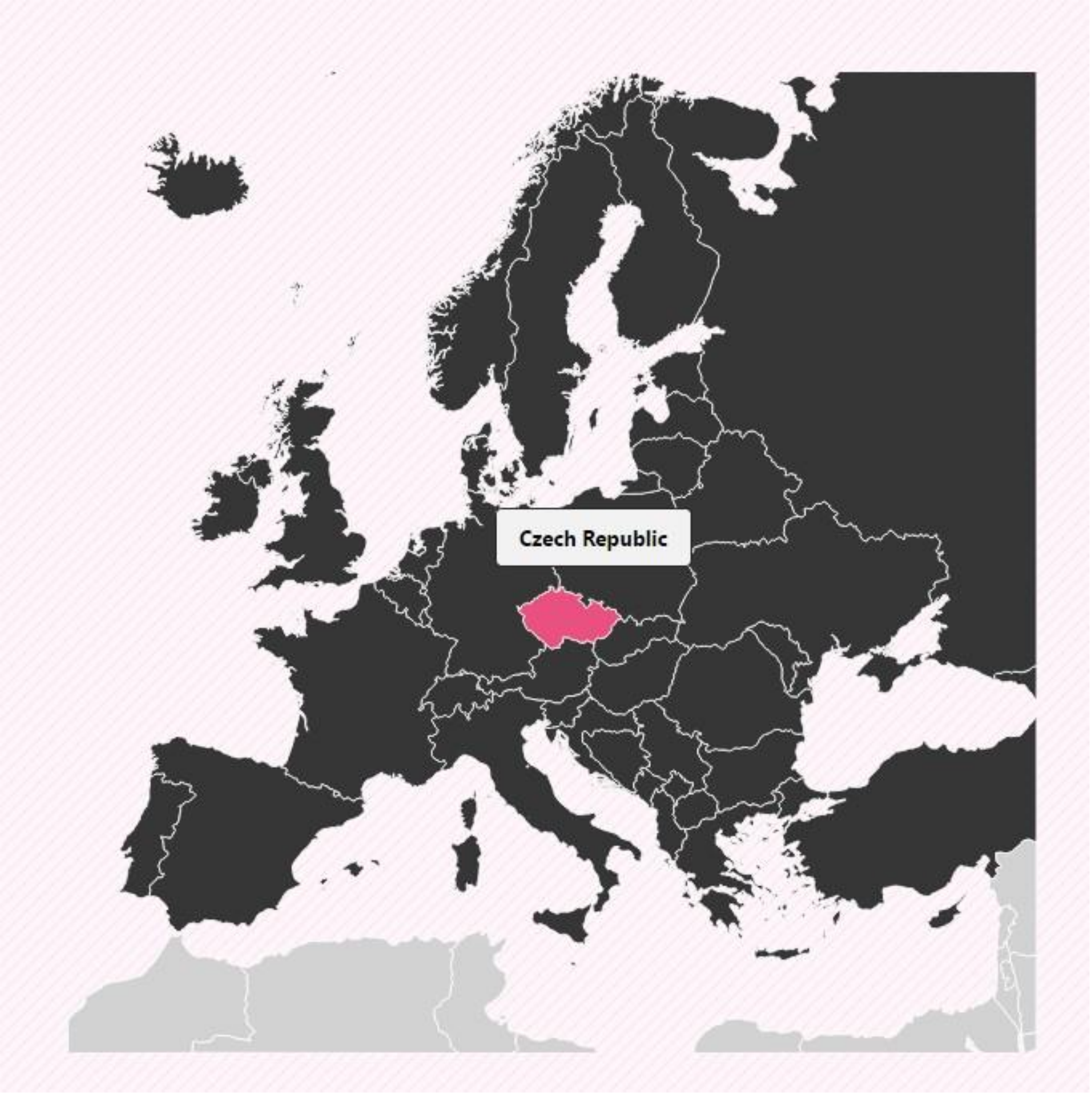
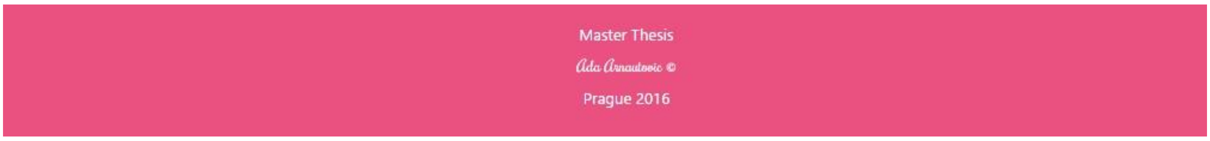


Figure 18 - Look of the map when Czech Republic is selected

The last part of the map part of the website, also being redirected as to a homepage, is the footer, that exists on each of the web pages being created. The footer is simple and basic, a single <div> having background color of the central European category – hue of the red, represented in a hexadecimal notation as a #ea5080, with only three headings stating the type of the project being worked on – Master thesis, name of the creator, and the place and date of creation – Prague 2016.

Figure 19 shows the look of footer with three headings it has.

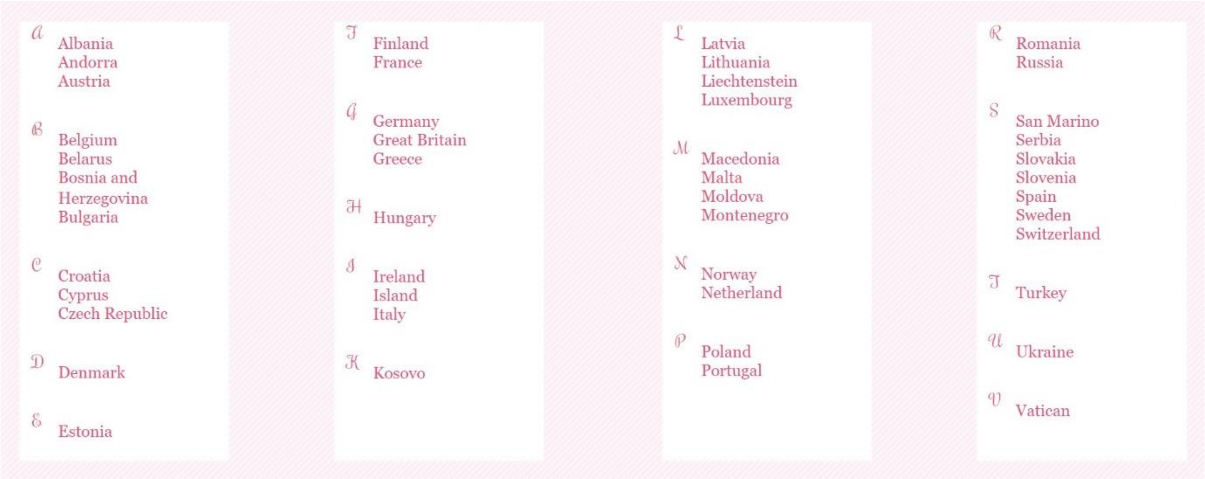


Master Thesis
Ada Arnautovic ©
Prague 2016

Figure 19 - Footer of the web page

4.2 Geography

Geography navigation tab serves to give an overview of all the countries in the Europe, in an alphabetical order, that provide links to each country.



A
Albania
Andorra
Austria

B
Belgium
Belarus
Bosnia and Herzegovina
Bulgaria

C
Croatia
Cyprus
Czech Republic

D
Denmark

E
Estonia

F
Finland
France

G
Germany
Great Britain
Greece

H
Hungary

I
Ireland
Island
Italy

K
Kosovo

L
Latvia
Lithuania
Liechtenstein
Luxembourg

M
Macedonia
Malta
Moldova
Montenegro

N
Norway
Netherland

P
Poland
Portugal

R
Romania
Russia

S
San Marino
Serbia
Slovakia
Slovenia
Spain
Sweden
Switzerland

T
Turkey

U
Ukraine

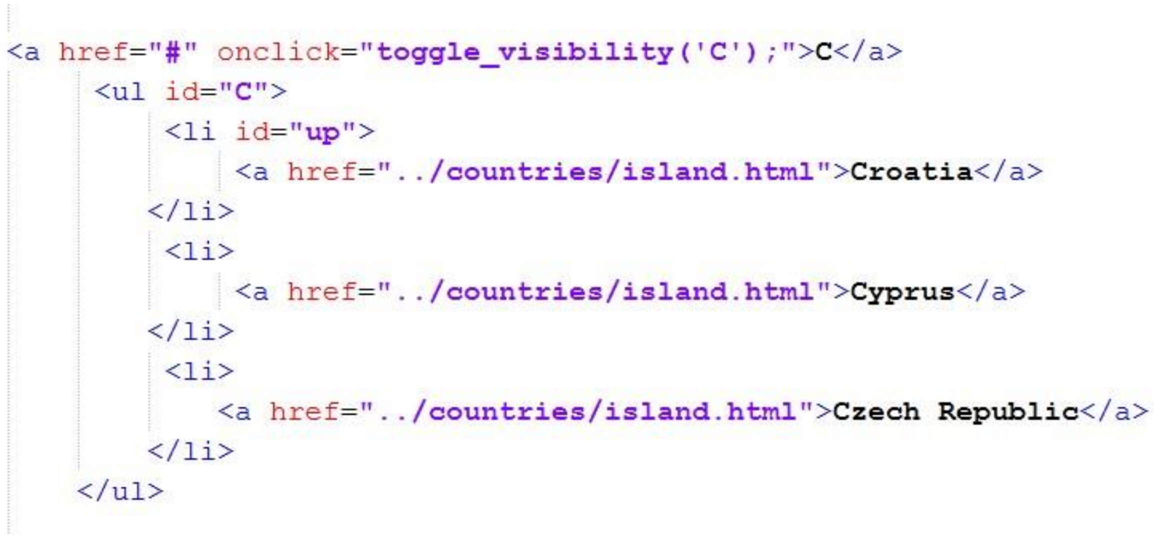
V
Vatican

Figure 20 - Geography tab of the website

Figure 20 shows the look of the geography tab of the page. It is basically an index off all the countries. Visibility of the countries can be toggled, on the mouse click on any of the letters of the alphabet that serve as category for the countries.

Figure 21 shows HTML code for the part of geography tab, the letter C. In this portion of code, unordered lists that contain the hyperlinks of all the countries whose name starts with "C" can be seen, as well as trigger of the toggle visibility ('C') function.

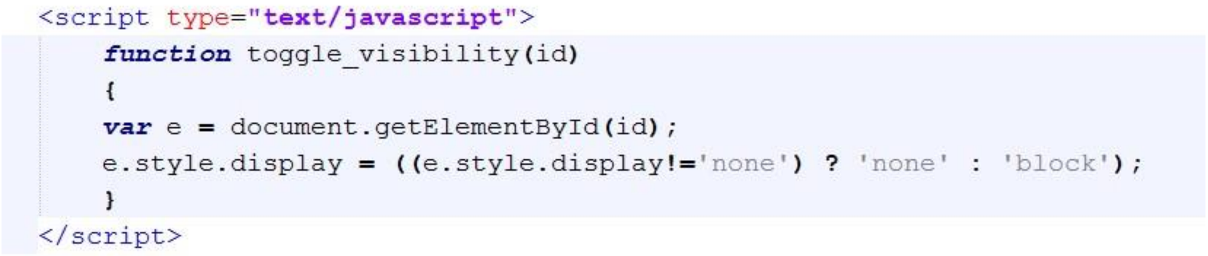
Figure 22 shows the JavaScript portion of code, where the function toggle visibility is created with parameter of id, which is later on called in HTML by inserting letter C as a parameter because it is an id in this specific case.



```
<a href="#" onclick="toggle_visibility('C');">C</a>
    <ul id="C">
        <li id="up">
            <a href="../countries/island.html">Croatia</a>
       </li>
        <li>
            <a href="../countries/island.html">Cyprus</a>
       </li>
        <li>
           <a href="../countries/island.html">Czech Republic</a>
       </li>
    </ul>
```

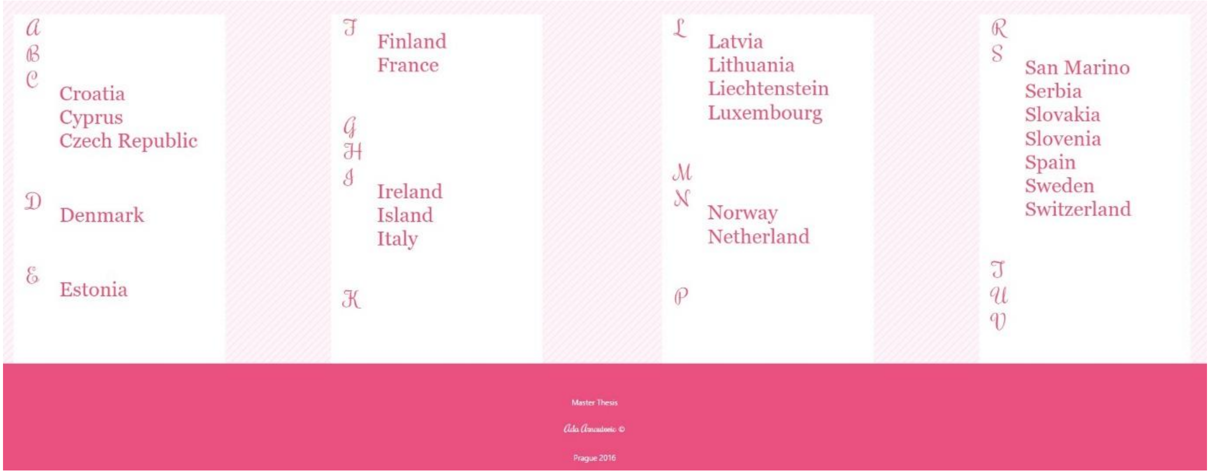
Figure 21 - Toggle visibility HTML

Moreover, the new variable, called e was created, that reads element from the document through its id, which is used to toggle the visibility and actually show and hide countries, every time their category is clicked. Another feature that can be chosen through CSS properties is to hide or show the content by default. One solution is to leave this code as it is, to show all the content before toggling the visibility, or adding display: none property if content should be not visible at first, i.e. by default.



```
<script type="text/javascript">
    function toggle_visibility(id)
    {
    var e = document.getElementById(id);
    e.style.display = ((e.style.display!='none') ? 'none' : 'block');
    }
</script>
```

Figure 22 - Toggle visibility JavaScript



A
B
C
Croatia
Cyprus
Czech Republic
D
Denmark
E
Estonia

F
Finland
France
G
H
I
Ireland
Island
Italy
K

L
Latvia
Lithuania
Liechtenstein
Luxembourg
M
N
Norway
Netherland
P

R
S
San Marino
Serbia
Slovakia
Slovenia
Spain
Sweden
Switzerland
T
U
V

Master Thesis
Ada Arnautovic ©
Prague 2016

Figure 23 - Geography tab when some categories are hidden

Figure 23 shows the look of the geography tab when some of the country categories already toggled the visibility, thus the whole content is not available for preview.

4.2.1 Czech Republic

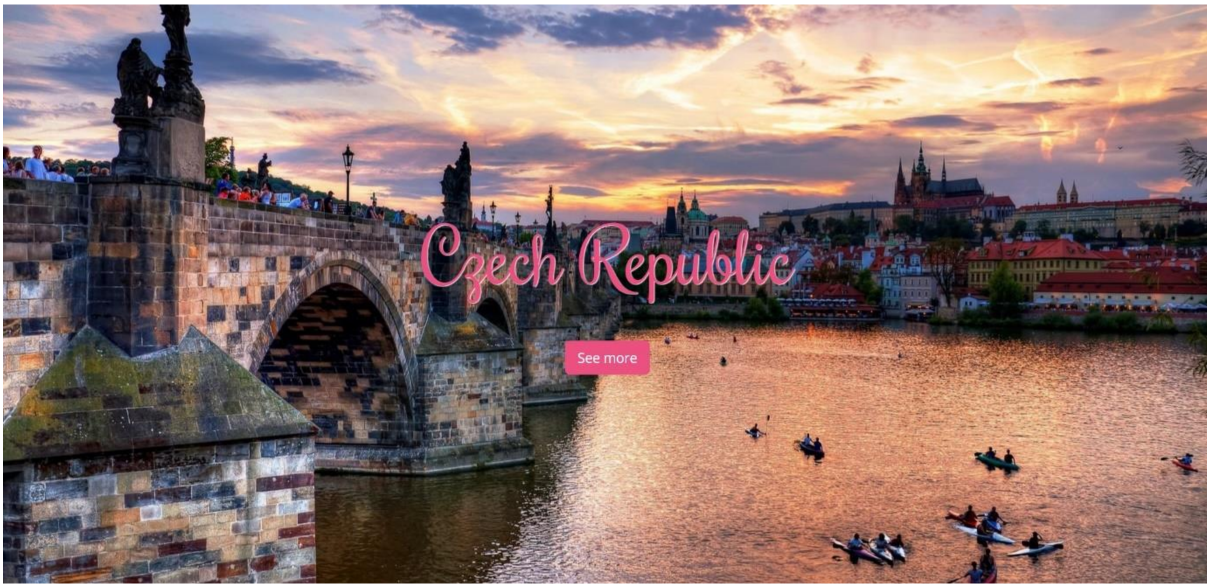
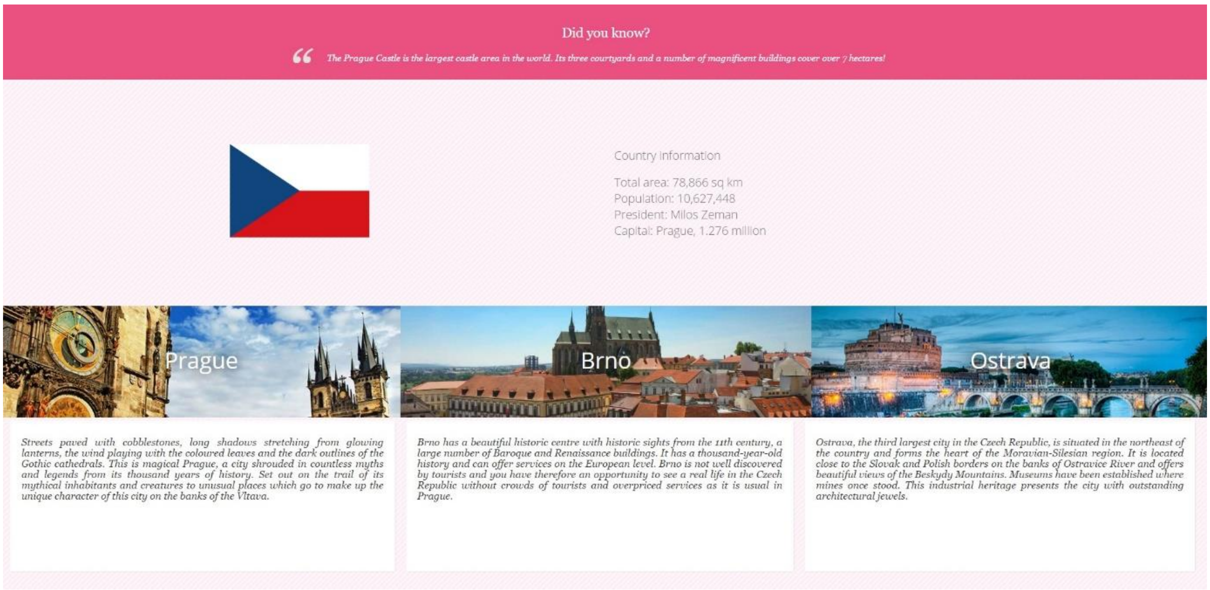


Figure 24 - Specific country part 1

Figure 24 represents the upper part of the specific country website, in this example, Czech Republic. It consists of the picture that has the width of 100%, name of the country in the heading, positioned in the middle and written using curly font Rochester that is saved in the external font document and linked to the original HTML document and "See more" button that redirects to the next part of the web page.



Did you know?

The Prague Castle is the largest castle area in the world. Its three courtyards and a number of magnificent buildings cover over 7 hectares!

Country information

Total area: 78,866 sq km
Population: 10,627,448
President: Milos Zeman
Capital: Prague, 1.276 million

Prague

Streets paved with cobblestones, long shadows stretching from glowing lanterns, the wind playing with the coloured leaves and the dark outlines of the Gothic cathedrals. This is magical Prague, a city shrouded in countless myths and legends from its thousand years of history. Set out on the trail of its mythical inhabitants and creatures to unusual places which go to make up the unique character of this city on the banks of the Vltava.

Brno

Brno has a beautiful historic centre with historic sights from the 11th century, a large number of Baroque and Renaissance buildings. It has a thousand-year-old history and can offer services on the European level. Brno is not well discovered by tourists and you have therefore an opportunity to see a real life in the Czech Republic without crowds of tourists and overpriced services as it is usual in Prague.

Ostrava

Ostrava, the third largest city in the Czech Republic, is situated in the northeast of the country and forms the heart of the Moravian-Silesian region. It is located close to the Slovak and Polish borders on the banks of Ostravice River and offers beautiful views of the Beskydy Mountains. Museums have been established where mines once stood. This industrial heritage presents the city with outstanding architectural jewels.

Figure 25 - Specific country part 2

Figure 25 shows the second part of the specific country website, and it consist of three main parts. First part is the small banner, that has "red" category color as a background, and its only content is a random, interesting "Did you know" fact. The second part is a <div> that has a country flag on the left side and the basic country information on the right side, including: total area in square meters, population, name of the president and name and population of the capital. The third part, has 3 different <div>s containing information about three biggest cities in the certain country. This information includes the famous landmark picture from each of the cities, positioned next to each other, followed by the short description of the city, placed in a three different columns on a white background, following a newspapers style.

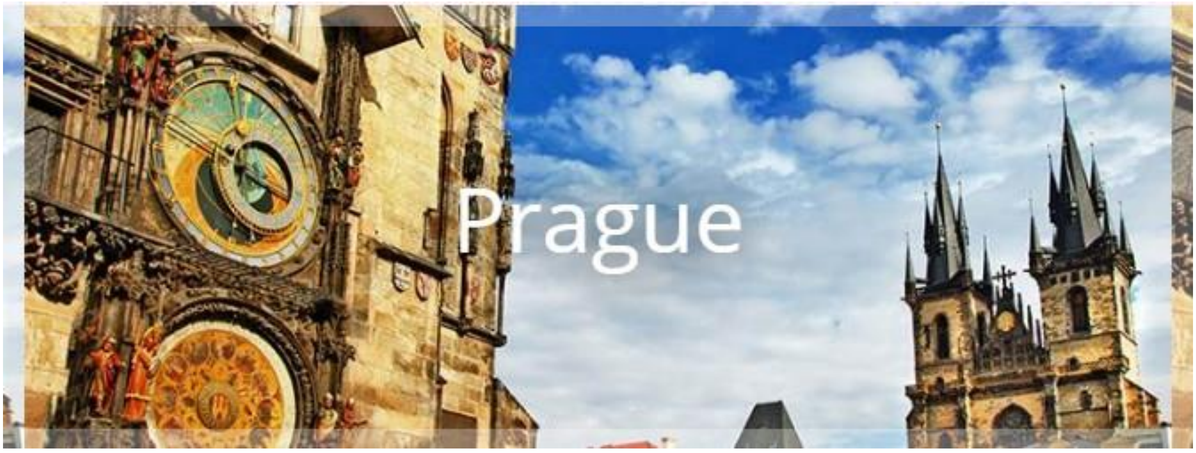


Figure 26 - Image while hovering

Figure 26 and figure 27 show the difference on the images while hovering over them (Figure 26) and the layout by default, without mouse interaction (Figure 27). This is done by manipulating CSS available from Bootstrap library and integrated through external CSS document. The same effect applies on all three pictures in a row, i.e. Prague, Brno and Ostrava. Basically, the illusion behind this effect is the opacity and transparency change, taking place on the whole borders along the picture.

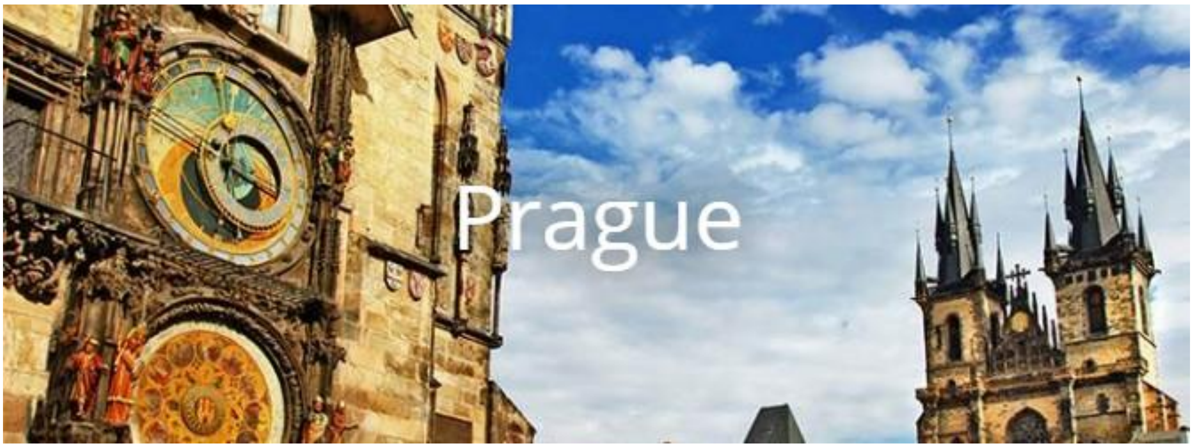
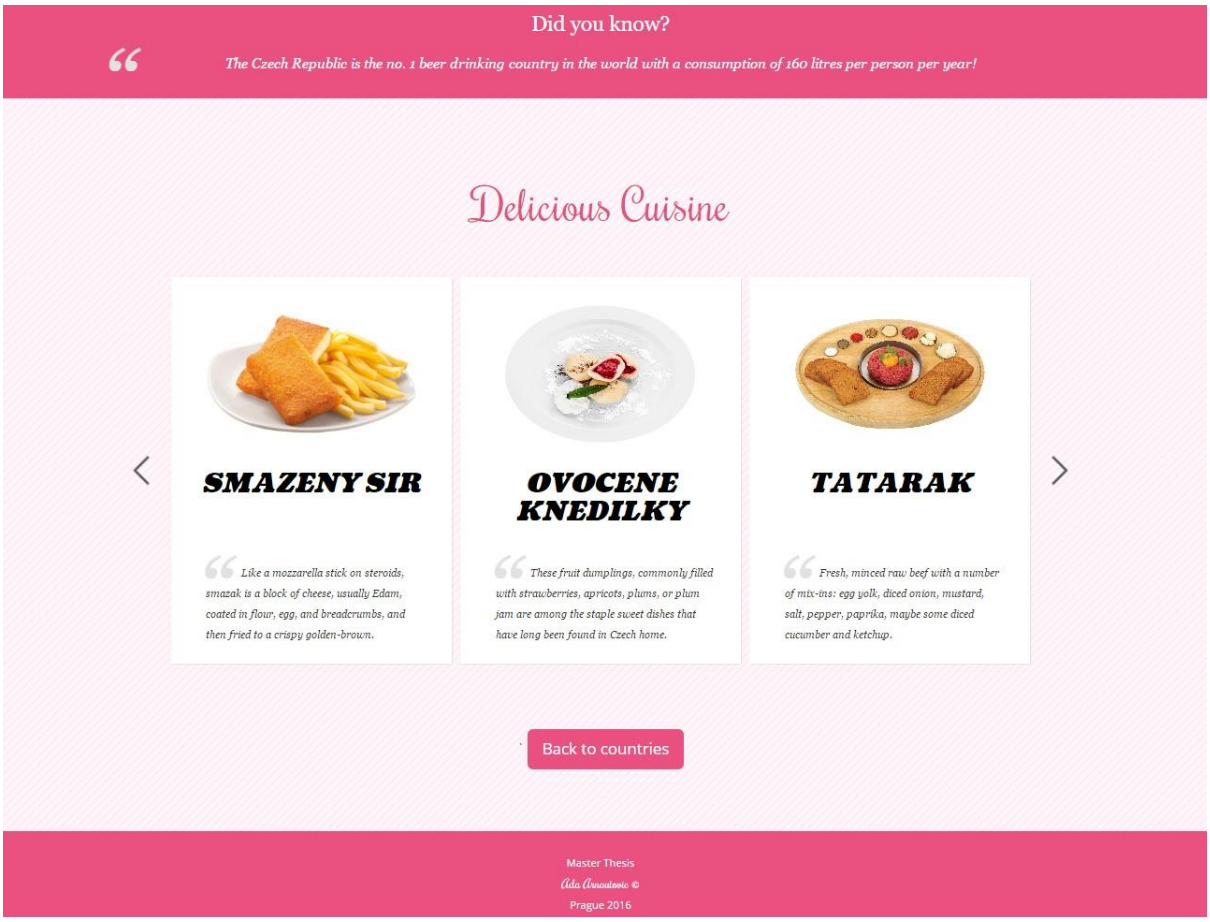


Figure 27 - Image without hovering

The third part of this specific country layout, as it can be seen from the figure 28 has one more banner with “Did you know” facts, that has the same design properties as the previous one. After the banner, there is a “Delicious Cuisine” part, that has a menu with the six dishes specific to the certain country. Each of them is positioned in the separate <div>, giving it a look as in the food menu. They contain the picture of the certain dish, name of the dish written in the Coustard font, that is linked externally to the HTML document, and the short description of the dish with quotation marks indented before the text.



The Czech Republic is the no. 1 beer drinking country in the world with a consumption of 160 litres per person per year!

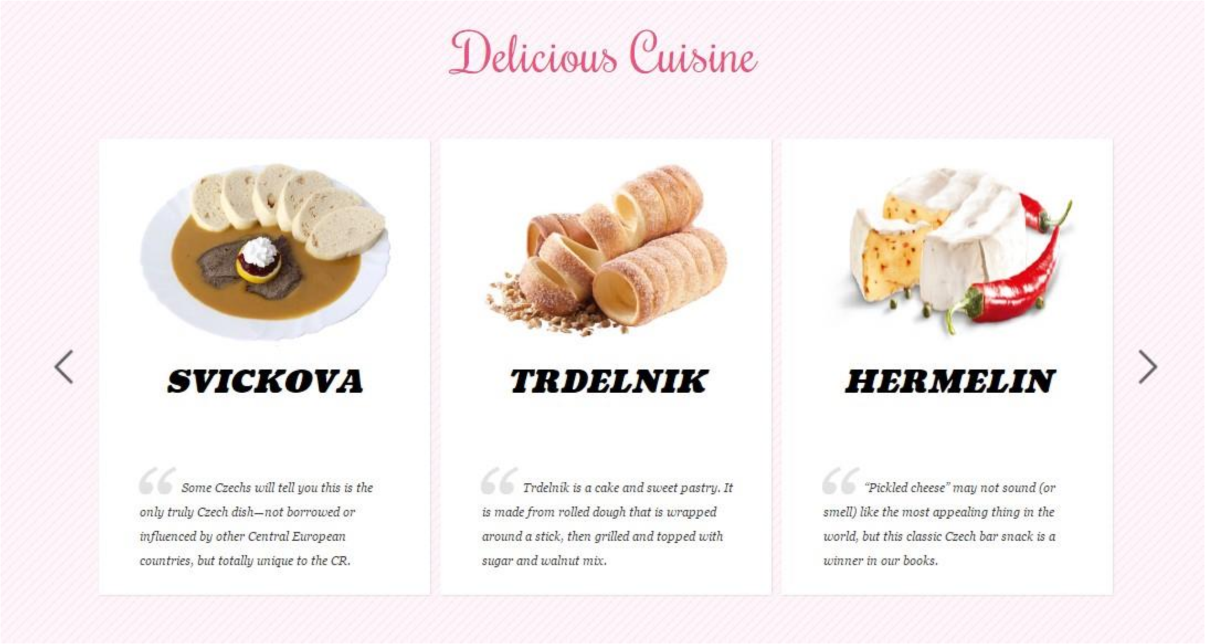
Like a mozzarella stick on steroids, smazak is a block of cheese, usually Edam, coated in flour, egg, and breadcrumbs, and then fried to a crispy golden-brown.

These fruit dumplings, commonly filled with strawberries, apricots, plums, or plum jam are among the staple sweet dishes that have long been found in Czech home.

Fresh, minced raw beef with a number of mix-ins: egg yolk, diced onion, mustard, salt, pepper, paprika, maybe some diced cucumber and ketchup.

Figure 28 - Specific country part 3

Figure 29 shows the remaining three items that cannot be seen in the figure 28 above. In the specific example of Czech Republic, chosen food is: fried cheese, dumplings stuffed with fruit, raw beef steak served with bread and various spices, beef in sour cream sauce served with dumplings, sweet pastry filled with chocolate that is usually sold on famous squares and streets and pickled cheese. Names were intentionally left in the native language, to keep the authenticity.



Delicious Cuisine

SVICKOVA

Some Czechs will tell you this is the only truly Czech dish—not borrowed or influenced by other Central European countries, but totally unique to the CR.

TRDELNIK

Trdelník is a cake and sweet pastry. It is made from rolled dough that is wrapped around a stick, then grilled and topped with sugar and walnut mix.

HERMELIN

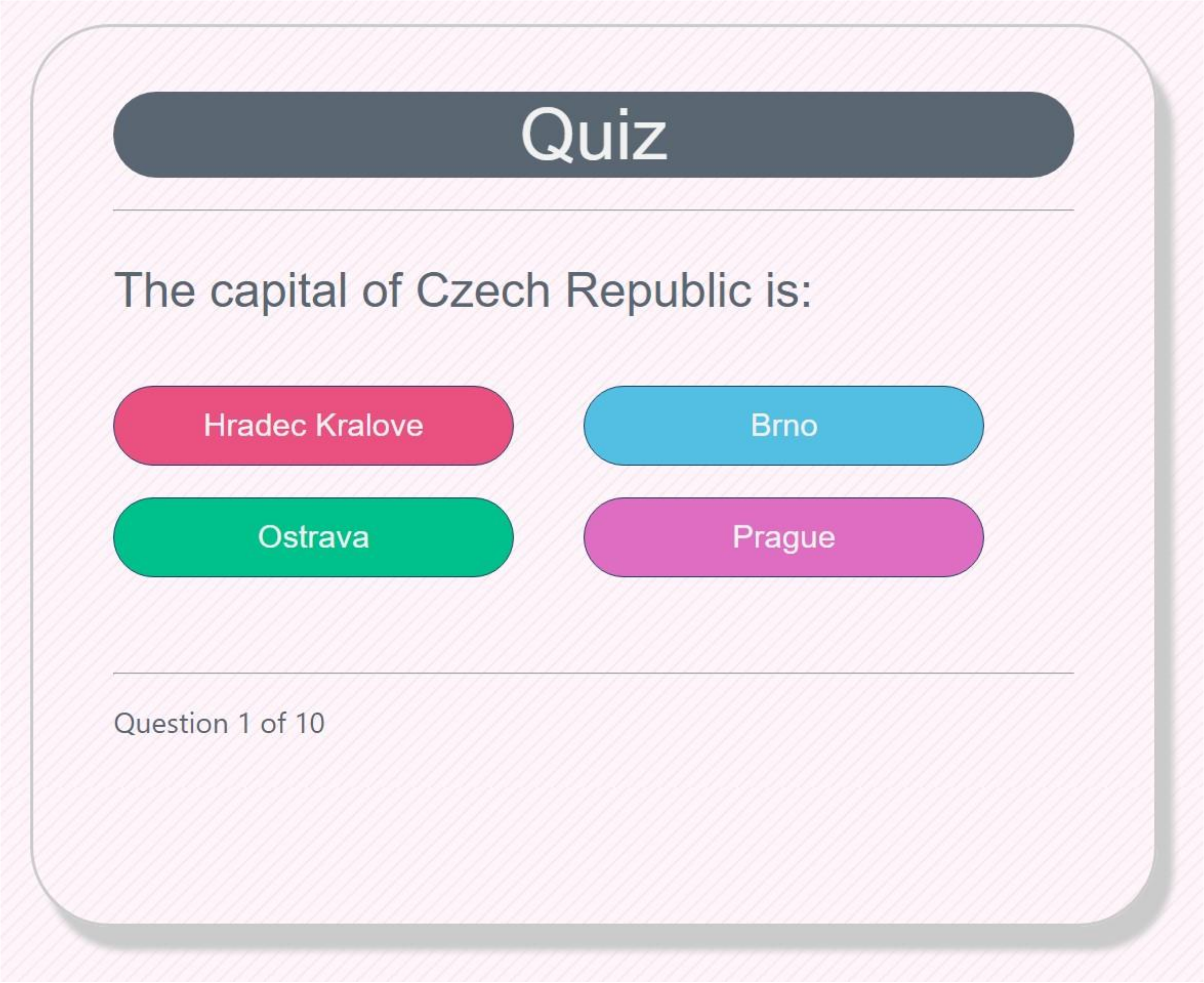
"Pickled cheese" may not sound (or smell) like the most appealing thing in the world, but this classic Czech bar snack is a winner in our books.

Figure 29 – Specific country part 3 variation 2

Last element that can be seen on the specific country page is "Back to countries" button that redirects the user to the geography tab where another country can be chosen, or anything else from the navigation menu. The whole CSS and JavaScript documents that explains the layout of this web page, including movement and scroll in the food menu can be found in the appendices 1 (CSS) and 3 (JavaScript).

4.3 Quiz

Third tile from the navigation menu directs the user to the "Quiz" page. Design and layout of the quiz is fairly simple. It is based on multiple-choice answers. Each question has four possible answers, three are wrong and one is correct. Four out of five primary colors are used when displaying the answers, while for the question and heading, neutral grey is used.

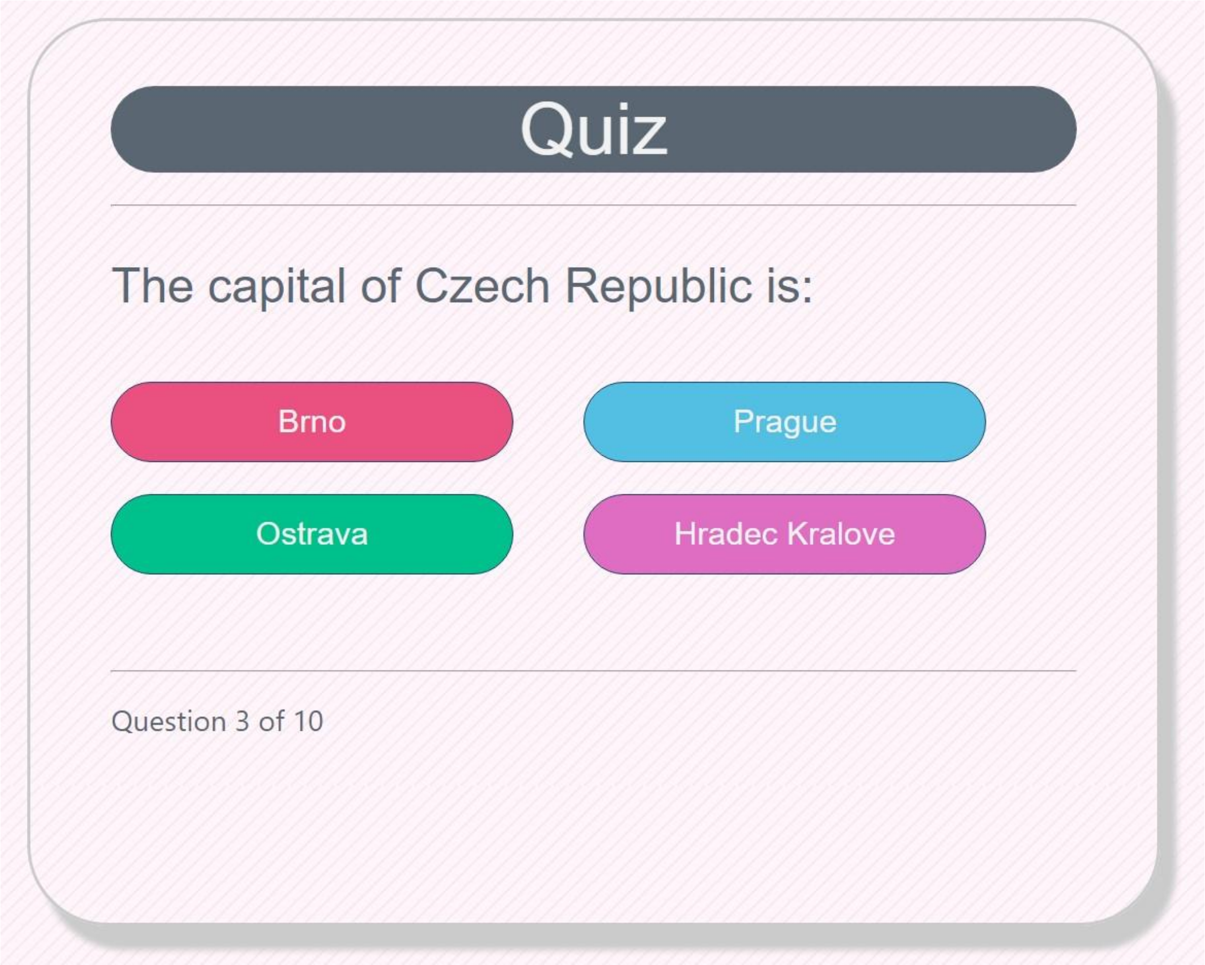


Quiz

The capital of Czech Republic is:

Figure 30 – Quiz layout

The quiz is created with three external JavaScript files. Function random is used in order for questions not to repeat in the same order. It applies also for the answers. The difference can be seen in the Figure 30 and Figure 31 – the same question came in different question order, and at the same time the answers were not distributed in the same order. In the Figure 30 it was the question number one and the correct answer was positioned in the lower right corner, in the violet box, while in the Figure 31, the same question had an ordinal number three, and Prague was placed in the blue box, in upper right corner.



Quiz

The capital of Czech Republic is:

Question 3 of 10

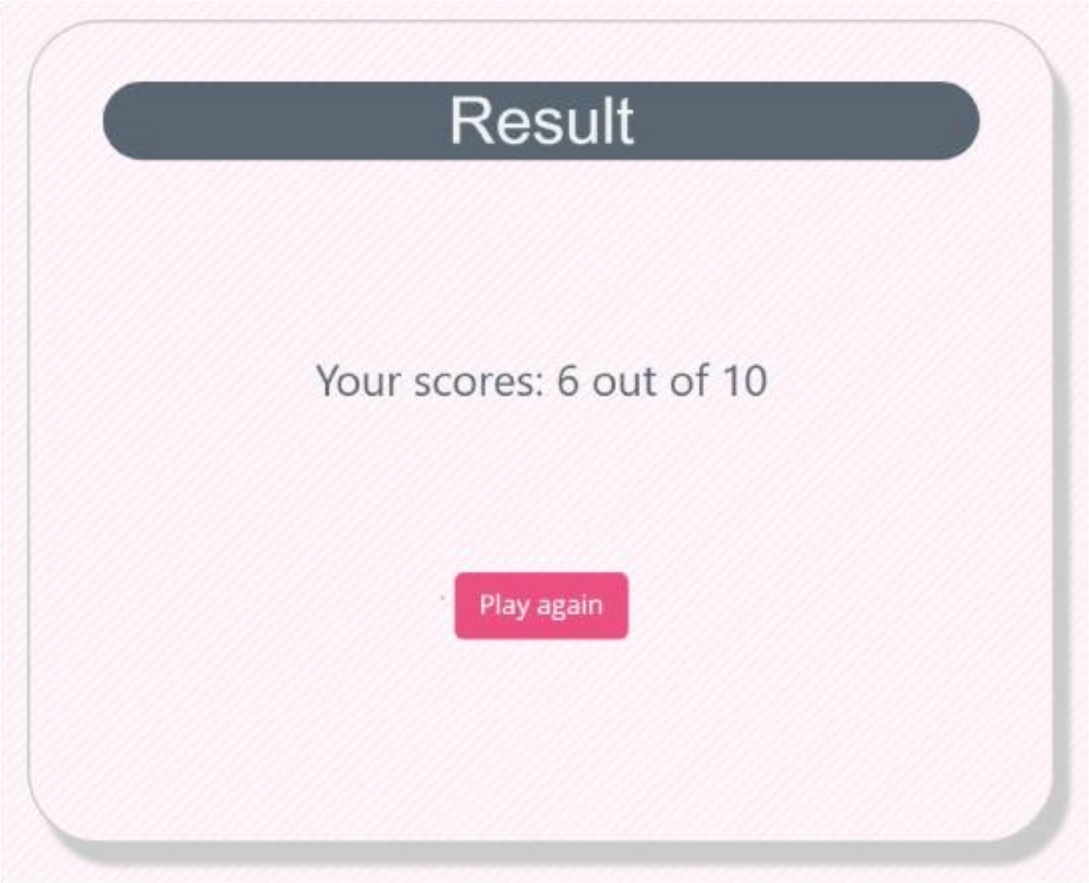
Figure 31 - Quiz layout shuffle

```
function Question (text, choices, answer) {
        this.text = text;
        this.choices = [].concat(choices).sort(function() {
        return Math.random() - Math.random();
  });
        this.answer = answer;
}
```

```
function Quiz (questions) {
        this.score = 0;
        this.questions = [].concat(questions).sort(function() {
        return Math.random() - Math.random();
  });
        this.QuestionIndex = 0;
}
```

Piece of code above shows two main functions, the Question function and the Quiz function. The Question function contains the text of the question, the possible choices and the answer (the right choice). It uses Math.random() function to randomize the order of the choices displayed below the question. The Quiz function initializes score and question index to be zero at the beginning and it also randomizes (again, using Math.random()) the order of the questions.

Once all ten questions are done, the "result" page is displayed, showing the total score that user got and offering a "play again" button, that refreshes the page and gives the user an option to redo the quiz as many times as user wants. Figure 32 shows the layout once the quiz is done.

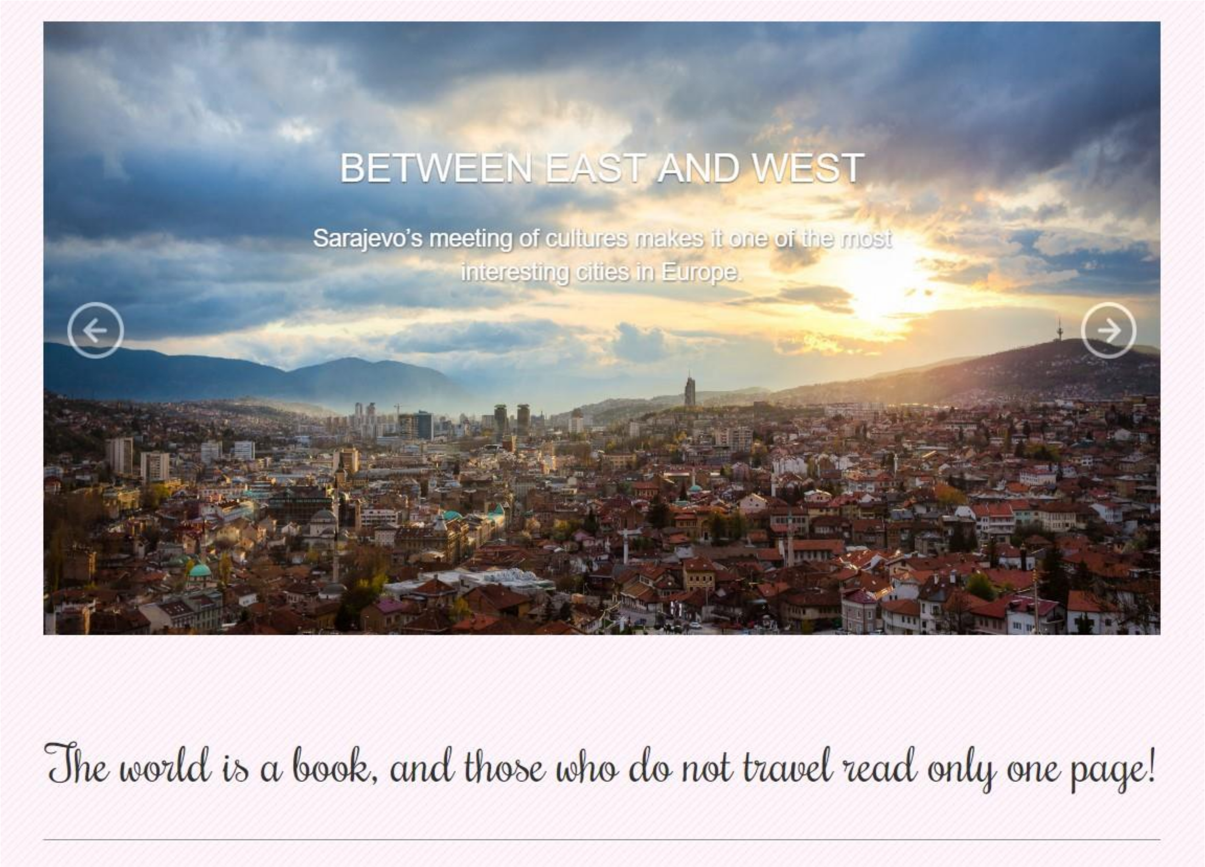


Your scores: 6 out of 10

Figure 32 - The Result Quiz layout

4.4 Travel

The fourth tile in the navigation menu is a travel tile. This is where the content from the business side, i.e. travel agency would appear. As the initial idea was to create an interactive map of Europe that would serve for three purposes – to provide education in Geographic field (Geography tab), to provide fun in the shape of simple game (Quiz tab) and to provide business oriented content – this is the last piece of puzzle.



BETWEEN EAST AND WEST

Sarajevo's meeting of cultures makes it one of the most interesting cities in Europe.

The world is a book, and those who do not travel read only one page!

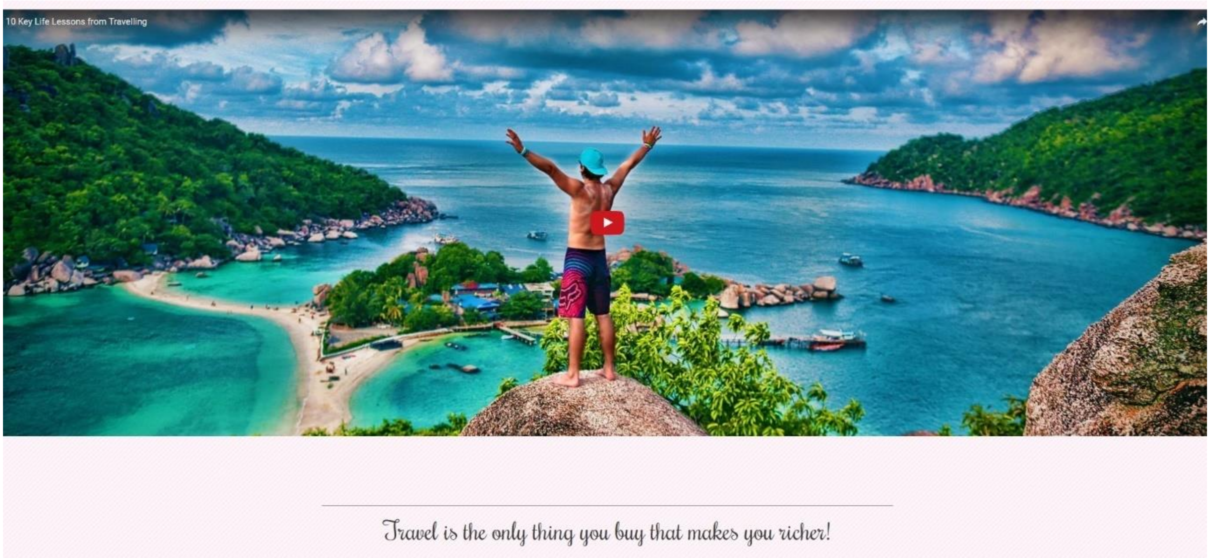
Figure 33 - Travel tab

Figure 33 shows the layout of the upper part of the travel tab. Design keeps being consistent, clear and minimalistic throughout the whole website, which can be seen from this tab as well – there is a curly Rochester font, panoramic-view picture and, of course, the background that is used on the other pages as well. Apart from that, there is an interactive picture gallery.

Pictures are being changed by default, as a slideshow, and the same effect can happen if a user presses these small arrows on the right and left side.

```
$('.carousel').carousel({
  interval: 3500, // in milliseconds
  pause: 'none' // set to 'true' to pause slider on mouse hover
})
```

The code sample below is used in the creation of this interactive picture gallery. Name of the class is “carousel” and ‘interval: 3500’ means that every 3500 milliseconds, the image will be replaced with the next one, while ‘pause: none’ means there should be no pause by user interaction in the animation; if it was ‘pause: true’ for example, instead of ‘pause: none’, the slider would be paused on the mouse hover.



10 Key Life Lessons from Travelling

Travel is the only thing you buy that makes you richer!

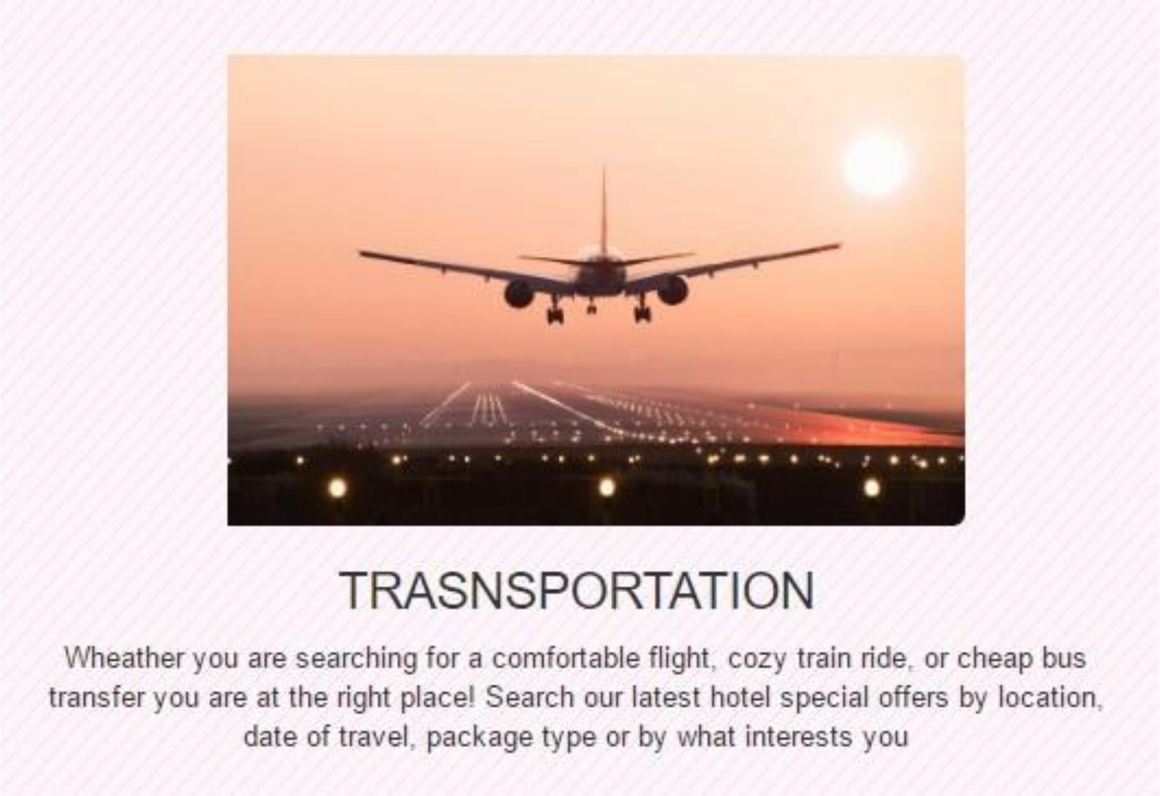
Figure 34 - Travel tab video

The second part of the travel tab, after interactive picture gallery is the video and one more quote related to travelling. It is known that videos evoke different emotions with the end users, then pictures and text do. They tend to be more convincing since they are (usually) very realistic and they include motion combined with the sound, that is proven to be appealing to the human mind. For that reason, it is always a good idea to include some moving content, in order for users to feel more connected with the topic itself.

Figure 34 is the layout of the video in the Travel tab. The name of the video is “Ten key life lessons from travelling” and it was posted on YouTube three months ago, on the “High on Life” channel.

The way to embed YouTube video in HTML is very simple, the main element is <iframe> and apart from that, all the other necessary HTML elements are basic and already explained in the previous part of this research. The code is as following:

```
<iframe
width="100%"
height="850"
src="https://www.youtube.com/embed/pu3ngqOxLbQ"
style="border:0;
scrolling=no;"
frameborder="0"
allowfullscreen>
</iframe>
```

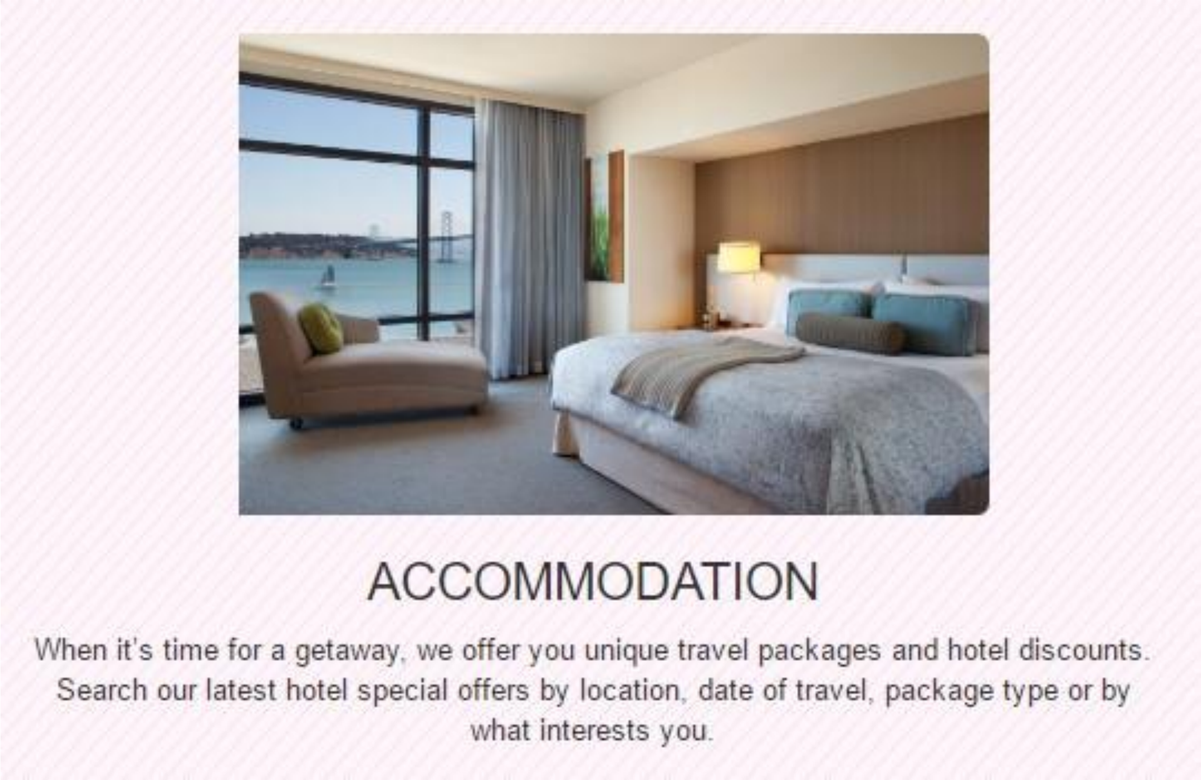


TRASNSPORTATION

Wheather you are searching for a comfortable flight, cozy train ride, or cheap bus transfer you are at the right place! Search our latest hotel special offers by location, date of travel, package type or by what interests you

Figure 35 - Transportation part of the Travel tab

Figure 35 and Figure 36 show the last part of the Travel tab, including transportation and accommodation.



ACCOMMODATION

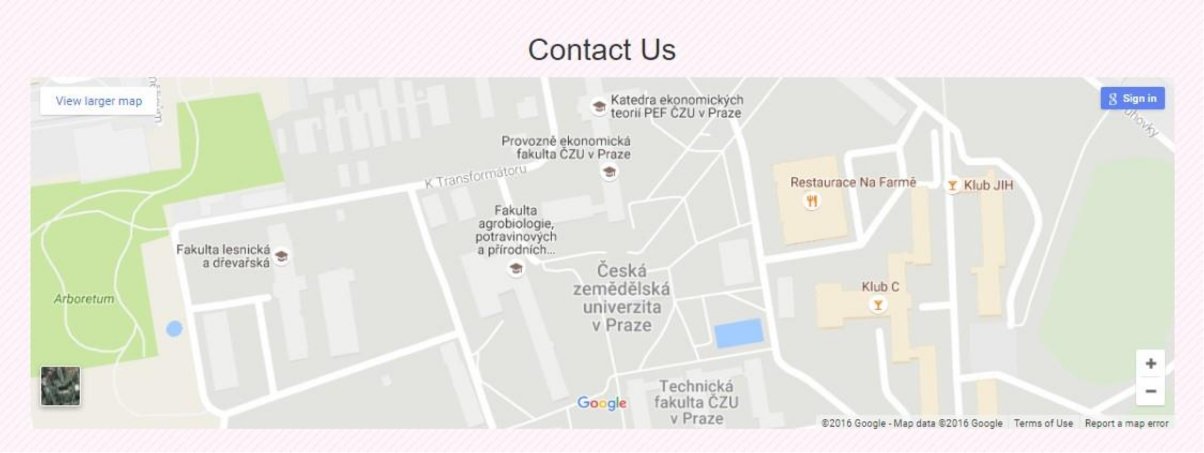
When it's time for a getaway, we offer you unique travel packages and hotel discounts. Search our latest hotel special offers by location, date of travel, package type or by what interests you.

Figure 36 - Accommodation part of the Travel tab

Accommodation and Transportation are left without being functional in this prototype of travel agency layout, meaning that in this specific page, they are pure images, without any hyperlinks opening new web pages or redirecting somewhere. Accommodation is meant to serve for hotel searches while transportation would be sort of an engine offering best deals when it comes to the flights, bus and train rides and other ways of transport. Since the purpose of this research is to test whether or not users would be more attracted to the interactive website offering the same content (i.e. selling journeys), the main focus was exactly on that part. The author believes that final results would not be any different if the Travel tab in this mock-up provided more details, in terms of specific destinations, dates of departure and similar. All of the countries (in this case, only European) that can be found on the map, have their own web pages, having same layout and consistent design, but unique content. Each of them can serve as a potential destination, or maybe the starting point, when it comes to the travel tab and real business.

4.5 Contact

Last, but definitely not least tile in the navigation menu is the contact tile. Unquestionably, very important part of any website. Once again, this research is conducted to test user's answers in the field of interactivity, comparing to many already-existing non-interactive contents, and there is no intention to make this website live in the near future; for that reason, there are no many specific contact details. Nonetheless, the style of a real website was maintained.



Contact Us

View larger map
Sign in
Katedra ekonomických teorií PEF ČZU v Praze
Provozně ekonomická fakulta ČZU v Praze
K Transformátoru
Restaurace Na Farmě
Klub JIH
Fakulta agrobiologie, potravinových a přírodních...
Fakulta lesnická a dřevařská
Česká zemědělská univerzita v Praze
Klub C
Arboretum
Technická fakulta ČZU v Praze
Google
©2016 Google - Map data ©2016 Google
Terms of Use
Report a map error

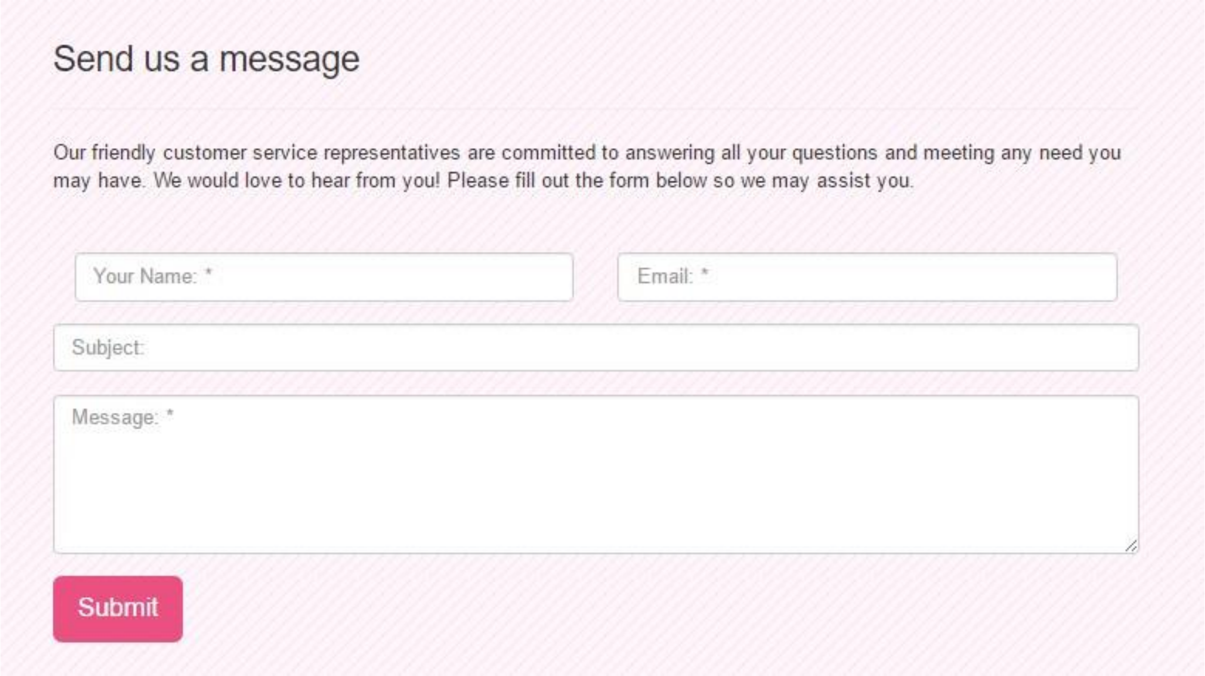
Figure 37 - Contact us - map

Figure 36 represents the upper part of the contact navigation tab. It is a map, or more precisely, a Google map, that can be zoomed in, zoomed out, moved across and everything else that any (Google) map can do. On page load, default coordinates show Czech University of Life Sciences, along with some of its faculties and other facilities: Faculty of Economics, Technical Faculty, Faculty of Agrobiology and Faculty of Forestry; club C, club JIH and restaurant On the Farm.

The way to integrate this map into already existing HTML code is as follows:

```
<iframe
width="100%"
height="350"
frameborder="0"
scrolling="no"
marginheight="0"
marginwidth="0"
src="https://www.google.com/maps/embed?pb=!1m14!1m12!1m3!1d1278.824174901862
3!2d14.373241832195257!3d50.13030187764026!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13
.1!5e0!3m2!1sen!2scz!4v1479831677733">
</iframe>
```

The code for the specific coordinates can be generated by the Google Maps, and then the only thing left to be done in order to have in on the website, is integrating it in <iframe> (as a one of the possible options), and setting some basic parameters, in order for the map to appear in a desired way. New CSS feature that was not explained in this research so far is ‘scrolling=”no”’. This is something that is used quite common, together with the iframes. It tells the assembler not to put either horizontal or vertical scroll bars on the right and bottom part of the screen; but instead to fit the content into chosen dimensions.

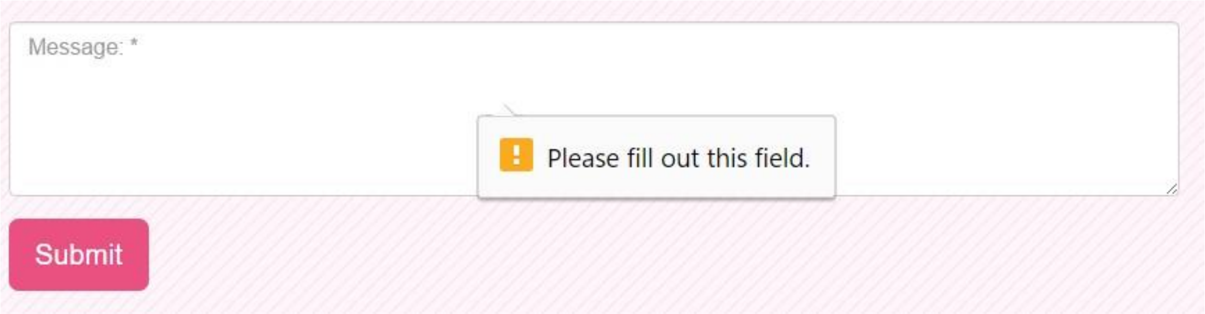


Send us a message

Our friendly customer service representatives are committed to answering all your questions and meeting any need you may have. We would love to hear from you! Please fill out the form below so we may assist you.

Figure 38 - Contact us forms

Apart from the map in the Contact tab, there is a bit more information on how to stay in touch. Figure 38 shows the bottom layout of the contact tab and it includes certain HTML forms and "Submit" button in order for user to send the message to the certain people (usually in a big company it would be Human Resource department, but in this specific case it is delivered to a website owner/developer/designer). The form has input fields for entering name, email, subject and message. The last form is a button to submit. The formed is designed to work in a way that all the fields except for the "subject" are mandatory. If a name field, or the content of the message are empty, the error will pop up asking to fill the field. Figure 39 is an example of that situation.


Message: *

Please fill out this field.

Figure 39 - Contact form with empty field

When it comes to the name and message content, the validator only checks if the fields are empty or not. Email validation is slightly different, since the validator checks the presence of the "@" symbol, and it requires a certain string to exist both, before and after commercial at.

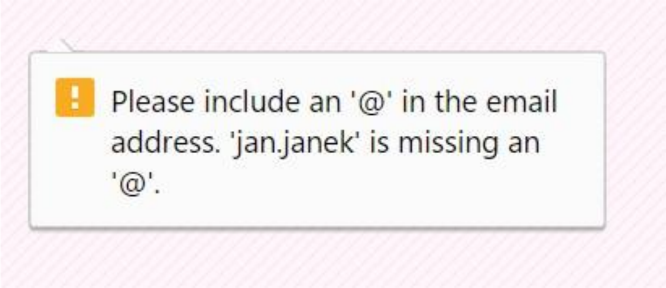
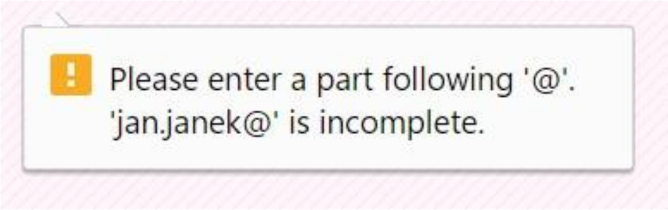

Please include an '@' in the email address. 'jan.janek' is missing an '@'.

Figure 40 - Incorrect email field error1

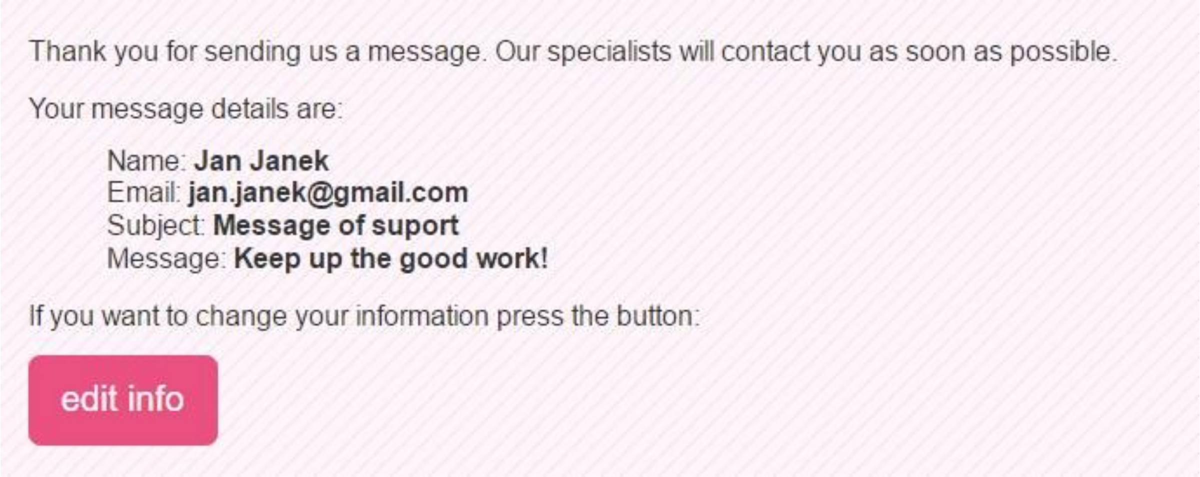
Figure 40 and Figure 41 show the error messages that appear in the form when email is not entered correctly according to the validator.



Please enter a part following '@'. 'jan.janek@' is incomplete.

Figure 41 - Incorrect email field Error2

Once all the fields are filled with the valid information and button to submit is clicked, the information that was entered is displayed with all the message details and a user have a possibility to edit any part of the information he wrote, by clicking on the “edit info” button. That action redirects back to the form, but the labels would not have the default values as they do before entering any information (i.e. “name”, “email”, “message” etc.); instead they would be pre-filled with the content that the user entered before pressing “edit info” button. Figure 42 serves as an example of the layout of that specific situation.



Thank you for sending us a message. Our specialists will contact you as soon as possible.

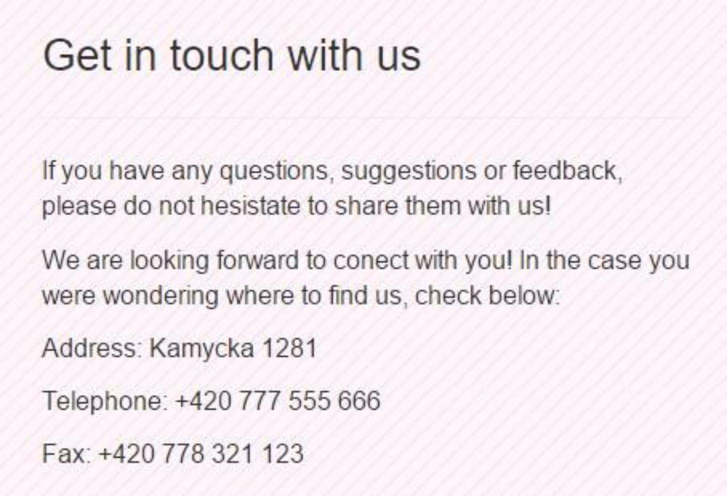
Your message details are:

Name: **Jan Janek**
Email: **jan.janek@gmail.com**
Subject: **Message of suport**
Message: **Keep up the good work!**

If you want to change your information press the button:

Figure 42 - Edit info from form

The very last part of the contact tab is essential contact information that is displayed to the user, which every website should consist of.



Get in touch with us

If you have any questions, suggestions or feedback, please do not hesitate to share them with us!

We are looking forward to conect with you! In the case you were wondering where to find us, check below:

Address: Kamycka 1281

Telephone: +420 777 555 666

Fax: +420 778 321 123

Figure 43 - Get in touch

Figure 43 shows the layout and (still simple, minimalistic and consistent) design of contact information available in the Contact tab.

5 Results and Discussion

After the website was created, an online survey using Google Forms was made in order to test whether the initial hypothesis that interactive media attracts more user than the interactive one, was true or not. Targeted population for this survey were final year master degree students at Czech University of Life Sciences, enrolled in English program. The population was estimated to be ~200. With the chosen confidence level of 90 as an amount of uncertainty that can be tolerated, and the margin of error up to 10% (expected outcomes should not be close to "50%-50%" approach, and for that reason the error up to even 10% can be tolerated, the recommended sample size was 50. Exactly one person more than recommended participated in this online survey. Nine questions were created, out of which five were based on a multiple-choices, two had linear scale from 1 to 10 to measure user's satisfaction and two were open questions with short answers expected. All the multiple-choice based questions and linear scale questions were required, while open questions were not obligatory to be answered.

List of the questions and the provided options for the answers is as follows:

1. How often do you use interactive media?
   a. On a daily basis
   b. Few times per week
   c. Few times per month
   d. Few times per year or less
2. Do you enjoy using interactive media more than non-interactive one?
   a. Definitely
   b. Slightly
   c. It is the same
   d. Not at all
3. Did you ever search for, and purchase a travel offer online?
   a. Yes, I do it frequently
   b. Yes, I did it few times
   c. Yes, I did it once

    d. No, I never did it

4. Do you like the concept of having everything at one place (entertainment, educational materials, business offers)?
    a. Yes, it makes a huge difference
    b. Yes, it is slightly more interesting
    c. I do not find some difference that matters
    d. No, I prefer more simple and specific content
5. Did you enjoy using this website?
    - The answer scale was from 1 to 10, 1 meaning “I didn’t enjoy at all” and 10 meaning “I enjoyed a lot”
6. Did you find it user-friendly and easy to navigate?
    - The answer scale was from 1 to 10, 1 meaning “It is not user-friendly at all” and 10 meaning “It is very user-friendly”
7. Would you more likely purchase a deal from a website like this than from the traditional one?
    a. Definitely from a website like this
    b. Probably from a website like this
    c. It wouldn’t influence my decision
    d. I prefer traditional ones
8. What was the thing you liked the most about this website?
    - Open question with “short text answer”
9. Was there something you did not like on this website? If yes, what?
    - Open question with “short text answer”

Figure 44 shows the summary of the results for the first and second question. 49% of participants answered that they use interactive media on a daily basis, 29.4% use it few times per week, 11.8% few times per month, while 9.8% of survey takers said they use it only few times per year or less. That was good statistics for the beginning, since it makes a total of 78.4% of the participants who use interactive media few times per week or more often, or 90.2% who use it few times per month or more often.

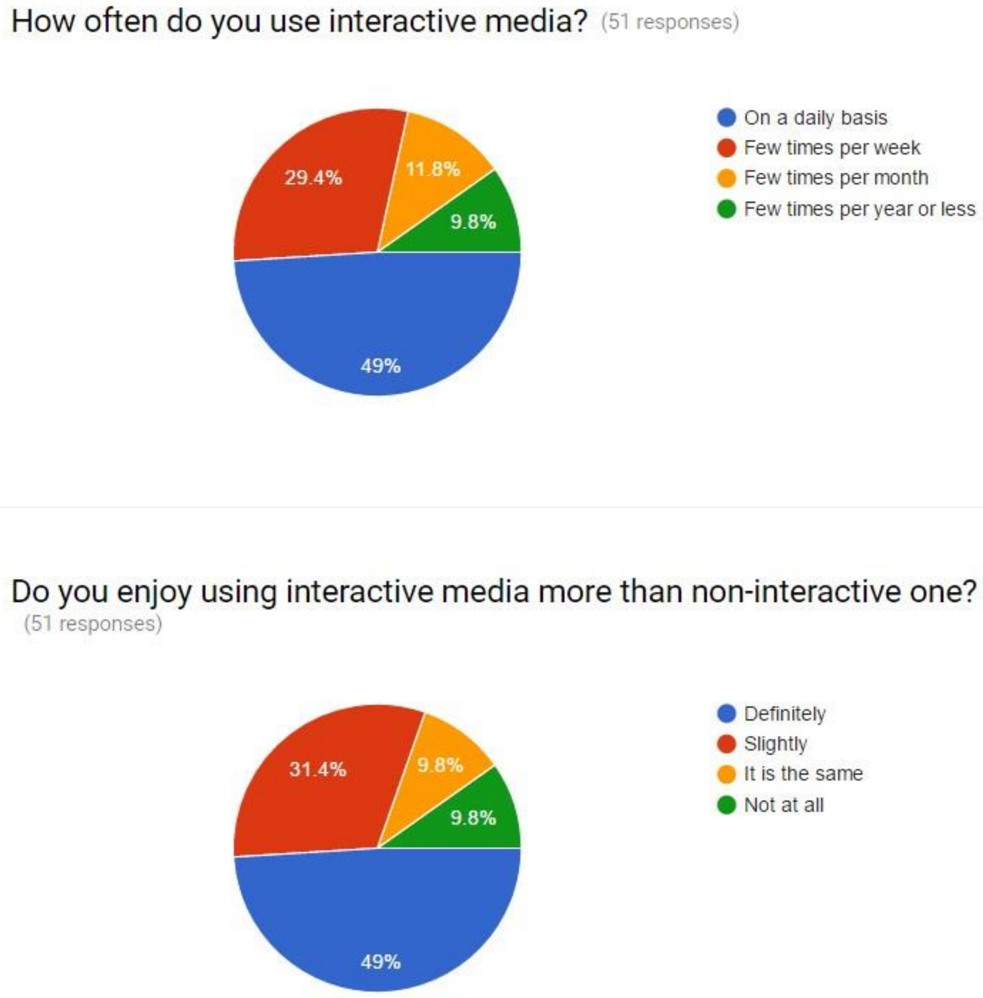


Figure 44 - Survey results 1

49% of surveyed final year master degree CULS students definitely enjoys using interactive media more than the non-interactive one. It is interesting to notice that exactly the same percentage of people are using it on a daily basis. 31.4% said that they slightly enjoy it, which makes the total of 81.4% of students who enjoy interactive media more. This was a very positive statistic for this research, which proves that interactivity do matter. 9.8% of participants said they it is the same for them, and the same percentage of survey takers said that they do not prefer it at all.

Figure 45 sums up the answers for the third and fourth question. According to the sample of students that did participate in the survey, 41.2% of them frequently do search for and purchase online travel offers. Only two percent less (39.2%) said that they did it a few times. 11.8% of the survey takes did it once, which sums up to 92.2% of positive responses. Remaining 7.8% never did it.

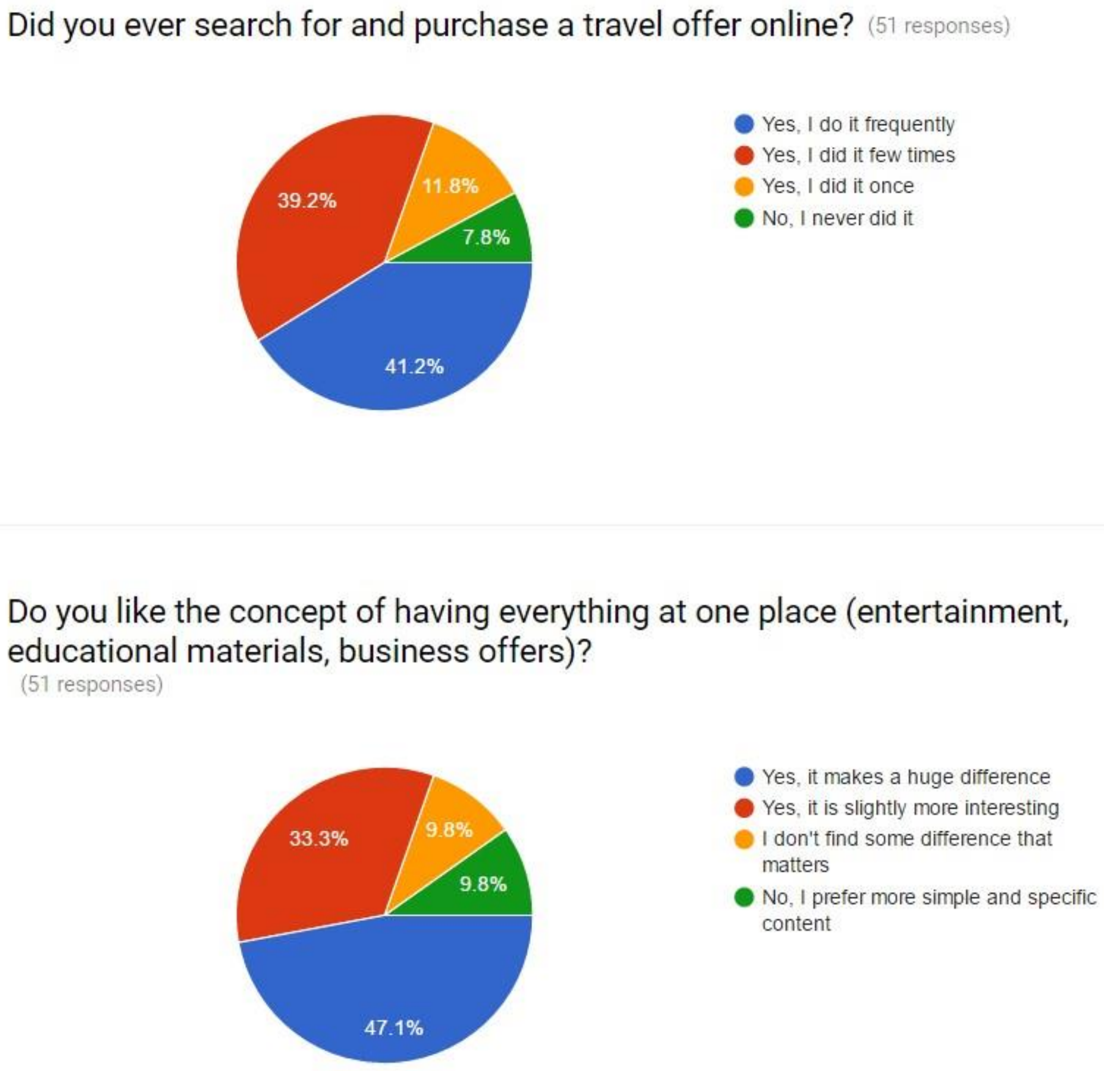


Figure 45 - Survey results 2

The question that is very important for the initial idea of this research – having the concept of entertainment, education and business at the one place got following answers: 47.1% of them likes it and say that it makes a huge difference; one third of the survey takers (33.3%) says that they like it because it makes it slightly more interesting, which makes a total of 80.4% participants who do like this concept. 9.8% of students does not find some difference that matters and the same amount of people prefer more specific and simple content.

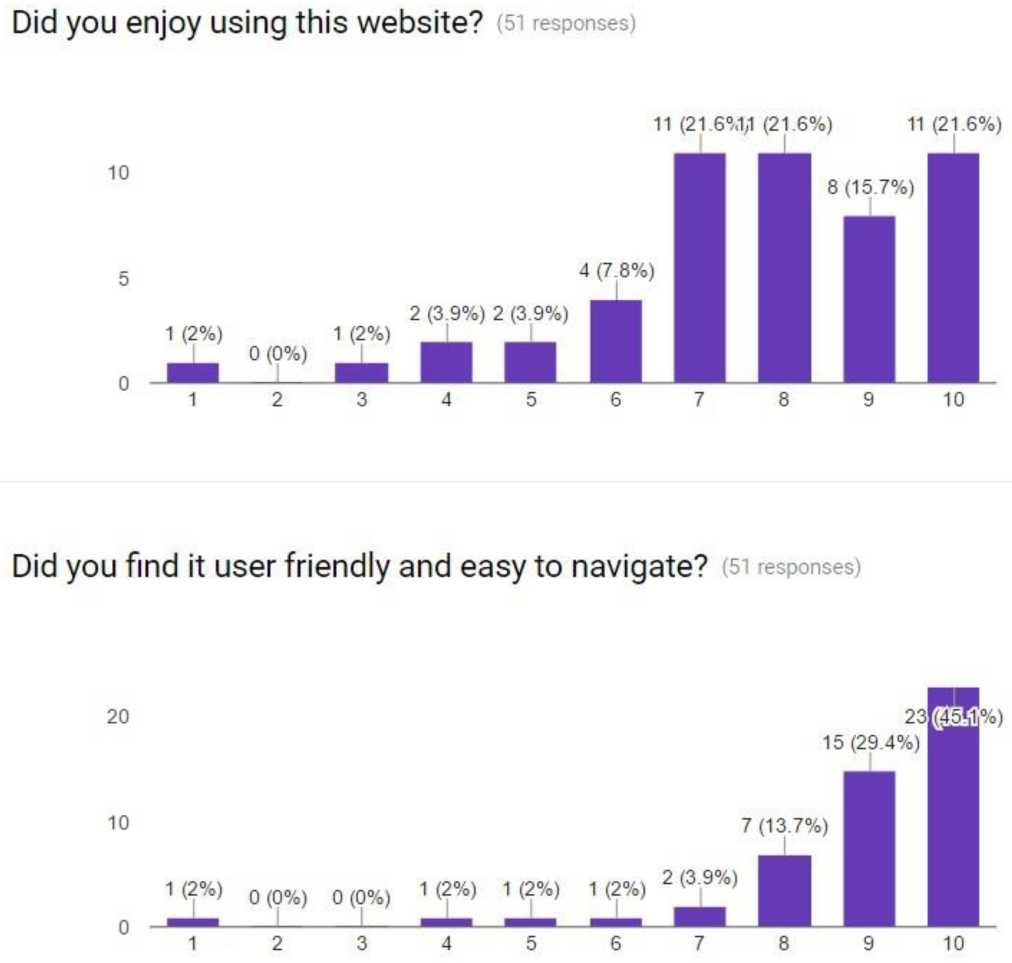


Figure 46 - Survey results 3

Figure 46 show the results from fifth and sixth questions. The questions had a linear scale for a possible answer, so the results are presented in a bar chart. When asked, did they enjoy using this specific website, 11 students (21.6%) enjoyed it a lot and gave it the highest scoring. The second highest scoring was chosen by 8 students (15.7%), while third and fourth best scorings were chosen by the same number of students as the best one – 11 (21.6% each). Scorings 6 and 5, that can be considered as a quite neutral answers in the middle were chosen by 4 (7.8%) and 2 (3.9%) participants. The lowest scoring, as well as the third lowest one was chosen by 1 student (2% each), the second lowest did not choose a single student, and the remaining 3.9% go to the 2 survey takers who gave the scoring of four. In total, these results can be considered as a very optimistic, since the 80.4% of CULS master degree students in a final year of English program gave a scoring of 7 and higher.

When they were asked if they consider a website user friendly and easy to navigate, the results were even better! 23 students (45.1%) gave the scoring of 10, meaning they do find it user friendly a lot; 15 of them (29.4%) gave the scoring of 9 and 7 of them (13.7%) gave a scoring of 8. Adding up these numbers, it is clear that even 88.2% of them gave an 8 or a higher scoring! Scoring of 7 was chosen by 2 students (3.9%), while scorings 1, 3, 4 and 6 were chosen by 1 student each (2% each). Scorings 2 and 3 were not chosen by a single student.

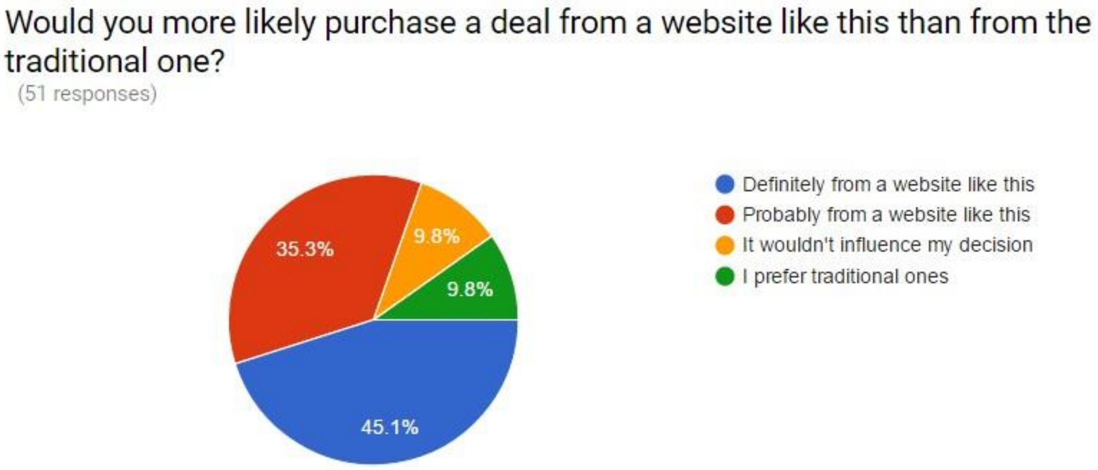
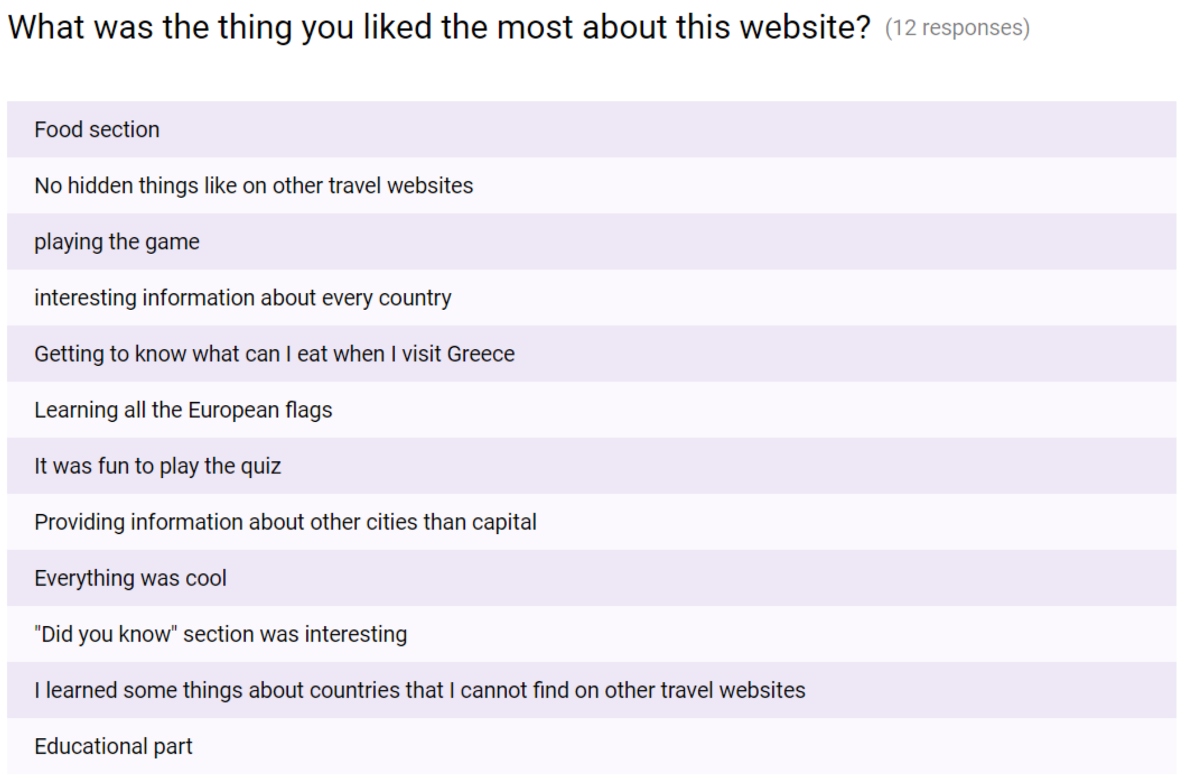


Figure 47 - Survey results 4

Figure 47 shows the distribution of the responses for the last multiple-choice based question. 45.1% of survey takers said that they would definitely purchase from a website like this before than from a traditional travelling website; 35.3% said that probably, website like this would attract them more when it comes to purchasing; 9.8% of students said that both, it would not influence their decision and that they prefer the traditional websites. This is, again, very bright statistic for this research, having in total 80.4% of students who give higher chances of purchasing to a website like one that is created by the author of this research.



What was the thing you liked the most about this website? (12 responses)

Food section

No hidden things like on other travel websites

playing the game

interesting information about every country

Getting to know what can I eat when I visit Greece

Learning all the European flags

It was fun to play the quiz

Providing information about other cities than capital

Everything was cool

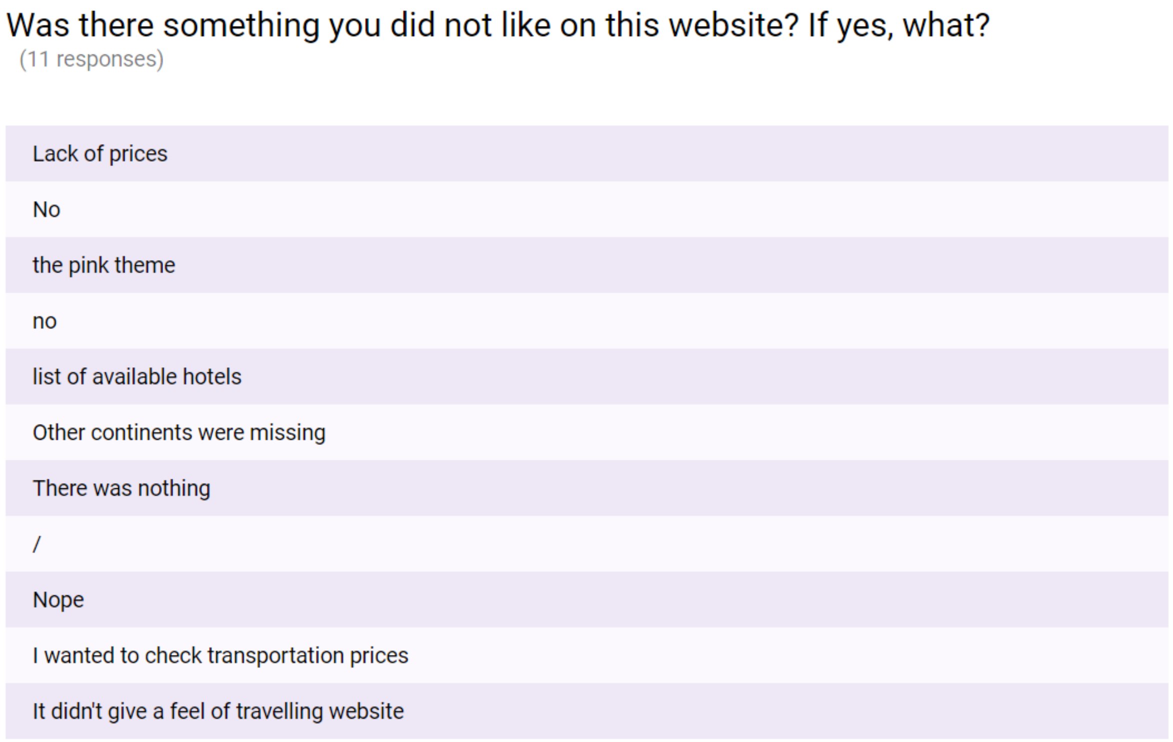
"Did you know" section was interesting

I learned some things about countries that I cannot find on other travel websites

Educational part

Figure 48 - Survey results 4

Figure 48 shows the responses for the first open question. Since it was not a required one, not all the users provided an answer. Nonetheless, 12 of them mentioned different things they liked the most about this website are. Most of the answers focus on the some of the tabs from the navigation menu. Majority of them liked the most the content on the Geography tab, with the answers “interesting information about every country”, “’Did you know’ section was interesting”, “learning all the European flags”, “providing information about other cities than the capital”, “I learned something about countries that I cannot find on other travel websites” and “educational part”. Interesting statistic was that two users (16.6%) when asked what they liked the most about the website, provided responses related to the food section. Taking these responses into consideration, and food section being part of each country’s profile, it can be concluded that the Geography tab was liked by the 8 out of 12 (66.6%) of the survey participants which means that two out of three people was happy that this section was here. That proves that it was the positive idea to include educational part in the concept of this website. Apart from this part, the only other answers that can be grouped and provide a significant statistics are the ones that were related to the quiz – “playing the game”, “it was fun to play the quiz”. This sums up that 16.6% of the participants (2 survey takers) were pleased to have the entertainment section. If these two sections would be grouped (educational and entertainment), it would be the favourite part of 83.33% (10 out of 12) of surveyed students.



Was there something you did not like on this website? If yes, what?
(11 responses)

Lack of prices

No

the pink theme

no

list of available hotels

Other continents were missing

There was nothing

/

Nope

I wanted to check transportation prices

It didn't give a feel of travelling website

Figure 49 - Survey results 5

Figure 49 is a summary of the answers to the last, ninth question: “Is there something you did not like about this website; if yes, what?” This was also the question that did not have required answer, and it was given response by 11 survey takes. Encouraging results are that 4 out of 11 (36.36%) replies were categorized as “No”. Two answers are about lack of prices being provided (18.18%), while other answers were quite unique and couldn’t be categorized in any specific group, including: “the pink theme”, “other continents were missing” and “list of available hotels”.

6 Conclusion

To conclude the research, after creating a website of interactive map of Europe, with approach of putting the same emphasis on the Educational, Entertainment and Business-oriented approaches, and testing whether the users would prefer it to the other available websites created for travelling, the results were affirmative. 80.4% of the targeted population; which were Czech University of Life Sciences final year master degree students who are attending program in English said they enjoyed using the website. Moreover, the vast majority of them prefers the concept and approach of a travelling website having other fields than commercial included. Perhaps the most important conclusion associated to the profit-side is that four out of five analysis contributors would have increased chances of purchasing from a website that has this approach.

7 References

1 The World Wide Web Consortium. Tim Berners Lee. 2016. [Online] June 30, 2016. [Cited October 03, 2016.] Available at: https://www.w3.org/People/Berners-Lee/

2 The birth of Web. Cern Switzerland. 2016. [Online] April 14, 2014. [Cited October 03, 2016.] Available at: http://home.cern/topics/birth-web

3 The history of web design. Hubspot. Lindsay Kolowich. [Online] March 09, 2015. [Cited: October 04, 2016.] Available at: http://blog.hubspot.com/marketing/web-design-history

4 Make an interactive website. Code Academy. 2016. [Online] [Cited: October 07, 2016.] Available at: https://www.codecademy.com/en/skills/make-an-interactive-website/topics/your-first-program/jquery-introduction

5 Difference between HTML and HTML5. Difference between. 2014-2016. [Online] October 10, 2016. [Cited: October 10, 2016.] Available at: http://www.differencebtw.com/difference-between-html-and-html5/

6 HTML Overview. Tutorials point. 2016. [Online] [Cited: October 10, 2016.] Available at: https://www.tutorialspoint.com/html/html_overview.htm

7 Dive into HTML5. Mark Pilgrim. [Online] March, 25, 2010. [Cited: October 10, 2016.] Available at: http://diveintohtml5.info/

8 HTML5 SVG. Tutorial Republic. 2016. [Online] [Cited: October 16, 2016.] Available at: http://www.tutorialrepublic.com/html-tutorial/html5-svg.php

9 Where is HTML5 video supported? Encoding.com. 2016. [Online] [Cited: October 16, 2016.] Available at: https://www.encoding.com/html5/

10 Adding sound. Thinkful. Ben Miller. 2016. [Online] [Cited: October 16, 2016.] Available at: https://www.thinkful.com/learn/html5-audio-tag-tutorial/

11 Differences between HTMl and HTML5. International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 5. T.N.Sharma, Priyanka Bhardwaj, Manish Bhardwaj. September 2012.

12 A technical introduction to XML. O'Reilly Media Inc. Norman Walsh. [Online] [Cited: October 18, 2016.] Available at: http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58

13 If you know nothing about HTMl, this is where you start. HTML Goodies. Joe Burns. 2016. [Online] [Cited: October, 18, 2016.] Available at: http://www.htmlgoodies.com/primers/html/article.php/3616146

14

15 Features that help you publish anything, anywhere. Custom design, CSS basics. Wordpress. 2016. [Online] [Cited: October 20, 2016.] Available at: https://en.support.wordpress.com/custom-design/css-basics/

16 CSS tutorial. W3Schools. 2016. [Online] [Cited: October 20, 2016] Available at: http://www.w3schools.com/css/css_howto.asp

17 What is the difference between CSS2 and CSS3. The major changes to CSS3. About tech. Jennifer Kyrnin. 2016. [Online] [Cited: October 20, 2016.] Available at: http://webdesign.about.com/od/css3/a/differences-css2-css3.htm

18 JavaScript language. Tutorials point. 2015. [Online] [Cited: October 20, 2016.] Available at: http://www.tutorialspoint.com/javascript/javascript_tutorial.pdf

19 JavaScript Enlightenment. First Edition, based on JavaScript 1.5, ECMA-262, Edition 3. Cody Lindley. [Online] [Cited: October 24, 2016.] Available at: http://www.javascriptenlightenment.com/

20 jQuery. jQuery write less, do more. 2016. [Online] [Cited: October 24, 2016.] available at: https://jquery.com/

21 About Bootstrap. Bootstrap. 2016. [Online] [Cited: October 24, 2016.] Available at: http://getbootstrap.com/about/

22 What is the Document Object Model? Philippe Le Hégaret, W3C, Lauren Wood, SoftQuad, Software Inc., WG Chair. Jonathan Robie, Texcel (for DOM Level 1) [Online] [Cited October 28, 2016] Available at: https://www.w3.org/TR/DOM-Level-2-Core/introduction.html

23 JavaScript HTML DOM. W3Schools. 2016. [Online] [Cited: October 28, 2016.] Available at: http://www.w3schools.com/js/js_htmldom.asp

24 The mobile web handbook. Peter-Paul Koch. Markus Seyfferth. Published by Smashing Magazine GmbH. 2014. ISBN 978-3-94454093-1

25 Why switch to XHTML? Web reference. Lee Underwood. 2016. [Online] [Cited October 28, 2016.] Available at: http://www.webreference.com/authoring/xhtml/index.html

26 Interactive Media – What is that? Who is involved? Elaine England and Andy Finney ATSF. 2011. Interactive Media UK.

8.1 Appendix 1

```
html, body {
	background: url(../images/pattern.png) repeat top left;
	background-color: hsl(329, 100%, 98%);}
.mapWrapper {
margin:0 auto;
position:relative;}
#map {
float:left;
clear:both;
width:930px;
height:590px;}
.point {
position:fixed;
display:none;
padding:10px 15px;
background:#f2f2f2;
font-size:14px;
font-weight:bold;
border:1px solid #262626;
-moz-border-radius: 4px; /* FF1-3.6 */
-webkit-border-radius: 4px; /* Saf3-4, iOS 1-3.2, Android <1.6 */
 border-radius: 4px 4px 4px 4px; /* Opera 10.5, IE9, Saf5, Chrome, FF4, iOS 4,
Android 2.1+ */
#text {
 width:500px; height: 380px; overflow: auto;}
.mapWrapper img {
float: left;
padding-right: 5px; padding-top: 2px;}
.mapWrapper ul{
margin-left: 30px;}
.carousel-caption {
	top: 50%;
	margin-top: -250px;
	font-size: 24px;
	color: #fff;
	text-transform: none;}
.carousel-caption h1 {
	font-size: 60px; line-height: 60px;
	text-transform: uppercase;
	font-weight: 500;
	color: #fff;}
.carousel-caption p {
	font-weight: 400;
	margin: 35px 0 20px 0;
	text-shadow: 1px 1px 3px #666666;}
.carousel-caption h1 span {
	color: #ffdd3c;
	border: 2px solid #fff;
	padding: 0 4px 4px;
	border-radius: 4px;}
.carousel-control {
	opacity: 1;
	position: absolute;
	top: 0%;
	margin-top: 0px;
	-moz-opacity: 1;}
```

```
.carousel-control span {
	position: absolute;
	top: 50%;
	margin-top: -70px;}
.carousel-control .carousel-control-left {
	background: url('../images/slider/prev.png') no-repeat;
	width: 57px;
	height: 57px;
	left: 25px;}
.carousel-control .carousel-control-right {
	background: url('../images/slider/next.png') no-repeat;
	width: 57px;
	height: 57px;
	right: 25px;}
.boxess .carousel-control .carousel-control-left {top: 50%;margin-top: -27px;}
.boxess .carousel-control .carousel-control-right {top: 50%; margin-top: -27px;}
.boxess .carousel-caption {top: 90px; margin-top: 0;}
.boxess {
	padding-top: 35px;
	background: url(../images/pattern.png) repeat top left;
	background-color: hsl(329, 100%, 98%);
	position: relative;
	top: 0;}
.boxess .carousel-caption h1 {
	font-size: 40px;
	line-height: 40px;}
footer .btn-md, .recent-posts .btn-md { margin-top: 0px;}
.main__middle__container .row ul, .main__middle__container .row ol {
	padding-left: 20px;}
.main__middle__container { width: 100%;}
.homepage {
	position: relative;
	margin-top: 50%;
	z-index: 111;
	background: url(../images/pattern.png) repeat top left;
	background-color: hsl(329, 100%, 98%);}
.row { padding: 45px 0;}
p.big-paragraph {
	font-size: 14px;
	text-transform: uppercase;
	padding-top: 15px;
	color: #e8bd10;}
.orange {
	color: #1ba39c;
	font-size: 14px;
	text-transform: none;
	padding: 0 0 10px 0;}
.black-text {
	color: #1e1e1e;}
.to_top {margin-top: -80px !important;}
.line__bg h3 {color: #fff;}
.line__bg p {color: #fff;}
.testimonials p {
	color: #fff;
	text-transform: none;
	font-size: 14px;
	line-height: 18px;}
.testimonials p small {
	font-size: 14px;
```

```
      color: rgba(255,255,255,0.8);
      font-weight: normal;}
.testimonials p.small-paragrapher {
      text-transform: uppercase;
      font-size: 14px;
      font-weight: 500;
      font-family: "Dosis", sans-serif;
      color: #eeeeee;}
img.full-width {width: 100%;}
.line__bg ol.carousel-indicators {
      bottom: -10px;
      padding-left: 0;}
.line__bg .carousel-indicators li {
      border-radius: 0px;
      border: none;
      margin: 0 3px;
      width: 10px;
      height: 11px;
      background: url('../images/slider/pager.png') no-repeat center center;}
.line__bg .carousel-indicators li.active, .line__bg .carousel-indicators li:hover
{background: url('../images/slider/pager_a.png') no-repeat center center;}
header{
    position: relative;
    width: 100%;
    min-height: auto;
    text-align: center;
    color: #fff;
    overflow:hidden;
    background-position: center;}
header .header-content {
    position: relative;
    width: 100%;
    padding: 100px 15px 70px;
    text-align: center;}
header .header-content  h1 {
    margin-top: 0;
    margin-bottom: 20px;
    font-size: 100px;
    font-weight: 300;
    font-family: 'Rochester', sans-serif; /* italic font */
    text-shadow:1px 1px 1px #fff;
      text-align:center;}
.container h1{
      font-family: 'Rochester', sans-serif; /* italic font */
      text-align:center;
      padding-bottom:25px;
      padding-top:25px;
      text-shadow:1px 1px 1px #fff;
      color: #ea5080;
      font-size: 50px;
    font-weight: 100;}
@media(min-width:768px) {
    header {min-height: 99%;}
    header .header-content {
        position: absolute; top: 50%;
        padding: 0 50px;
 }
.section-heading { margin-top: 0; margin-bottom: 20px;}
```

8.2 Appendix 2

```
<?xml version="1.0" encoding="utf-8"?>
<data>
<mapSettings
mapWidth = "646"
mapHeight = "654"
offColor = "d2d2d2"
strokeColor = "ffffff"
useSideText = "false"
textAreaWidth = "300"
textAreaPadding = "100"/>
<defaultSideText>  <![CDATA[]]></defaultSideText>
<stateData   stateName = "Island" url = "../Countries/countries/island.html"
stateMode = "ON" initialStateColor = "363636" stateOverColor = "00c08b"
stateSelectedColor = "90ee90"><![CDATA[]]> </stateData>
<stateData stateName="Ireland" url = "../Countries/countries/ireland.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "53bfe2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Great Britain" url =
"../Countries/countries/greatbritain.html" stateMode = "ON" initialStateColor=
"363636" stateOverColor = "53bfe2" stateSelectedColor = "feb41c"><![CDATA[]]>
</stateData>
<stateData stateName="Malta" url = "../Countries/countries/malta.html" stateMode =
"ON" initialStateColor= "363636" stateOverColor = "df6dc2" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="San Marino" url = "../Countries/countries/sanmarino.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "df6dc2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Vatican" url = "../Countries/countries/vatican.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "df6dc2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Belgium" url = "../Countries/countries/belgium.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "53bfe2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Luxembourg" url = "../Countries/countries/luxembourg.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "53bfe2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Portugal" url = "../Countries/countries/portugal.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "df6dc2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Norway" url = "../Countries/countries/norway.html" stateMode
= "ON" initialStateColor= "363636" stateOverColor = "00c08b" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Serbia" url = "../Countries/countries/serbia.html" stateMode
= "ON" initialStateColor= "363636" stateOverColor = "df6dc2" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Kosovo" url = "../Countries/countries/kosovo.html" stateMode
= "ON" initialStateColor= "363636" stateOverColor = "df6dc2" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Sweden" url = "../Countries/countries/sweden.html" stateMode
= "ON" initialStateColor= "363636" stateOverColor = "00c08b" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Andorra" url = "../Countries/countries/andorra.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "df6dc2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
```

```
<stateData stateName="Albania" url = "../Countries/countries/albania.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "df6dc2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Bosnia and Herezegovina" url =
"../Countries/countries/bosnia.html"  stateMode = "ON" initialStateColor= "363636"
stateOverColor = "df6dc2" stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Croatia" url = "../Countries/countries/croatia.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "df6dc2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Macedonia" url = "../Countries/countries/macedonia.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "df6dc2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Bulgaria" url = "../Countries/countries/bulgaria.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "f8c54d"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Cyprus" uurl = "../Countries/countries/cyprus.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "df6dc2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Turkey" url = "../Countries/countries/turkey.html" stateMode
= "ON" initialStateColor= "363636" stateOverColor = "df6dc2" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Austria" url = "../Countries/countries/austria.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "ea5080"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Czech Republic" url =
"../Countries/countries/czechrepublic.html" stateMode = "ON" initialStateColor=
"363636" stateOverColor = "ea5080" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Denmark" url = "../Countries/countries/denmark.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "00c08b"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Hungary" url = "../Countries/countries/hungary.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "ea5080"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Poland" url = "../Countries/countries/poland.html" stateMode
= "ON" initialStateColor= "363636" stateOverColor = "ea5080" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Slovakia" url = "../Countries/countries/slovakia.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "ea5080"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Slovenia" url = "../Countries/countries/slovenia.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "ea5080"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Germany" url = "../Countries/countries/germany.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "53bfe2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Liechtenstein" url =
"../Countries/countries/liechtenstein.html" stateMode = "ON" initialStateColor=
"363636" stateOverColor = "53bfe2" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Switzerland"  url =
"../Countries/countries/switzerland.html" stateMode = "ON" initialStateColor=
"363636" stateOverColor = "53bfe2" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Belarus" url = "../Countries/countries/belarus.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "f8c54d"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
```

```
<stateData stateName="Estonia" url =
"../Countries/countries/estonia.html"stateMode = "ON" initialStateColor= "363636"
stateOverColor = "00c08b" stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Latvia" url = "../Countries/countries/latvia.html" stateMode
= "ON" initialStateColor= "363636" stateOverColor = "f8c54d" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Lithuania" url = "../Countries/countries/lithuania.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "f8c54d"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Moldova" url = "../Countries/countries/moldova.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "f8c54d"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Romania" url = "../Countries/countries/romania.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "f8c54d"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Ukraine" url = "../Countries/countries/ukraine.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "f8c54d"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Montenegro" url = "../Countries/countries/montenegro.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "df6dc2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Netherland" url = "../Countries/countries/netherland.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "53bfe2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="Italy" url = "../Countries/countries/italy.html" stateMode =
"ON" initialStateColor= "363636" stateOverColor = "df6dc2" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Greece" url = "../Countries/countries/greece.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "df6dc2"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
<stateData stateName="France" url = "../Countries/countries/france.html" stateMode
= "ON" initialStateColor= "363636" stateOverColor = "df6dc2" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Russia" url = "../Countries/countries/russia.html" stateMode
= "ON" initialStateColor= "363636" stateOverColor = "f8c54d" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Spain" url = "../Countries/countries/spain.html" stateMode =
"ON" initialStateColor= "363636" stateOverColor = "df6dc2" stateSelectedColor =
"feb41c"><![CDATA[]]></stateData>
<stateData stateName="Finland" url = "../Countries/countries/filand.html"
stateMode = "ON" initialStateColor= "363636" stateOverColor = "00c08b"
stateSelectedColor = "feb41c"><![CDATA[]]></stateData>
</data>
```

8.3 Appendix 3

```
function Question (text, choices, answer) {
       this.text = text;
       this.choices = [].concat(choices).sort(function() {
       return Math.random() - Math.random(); });
       this.answer = answer;}
Question.prototype.correctAnswer = function (choice) {
       return choice === this.answer;
} function Quiz (questions) {
       this.score = 0;
       this.questions = [].concat(questions).sort(function() {
       return Math.random() - Math.random(); });
       this.QuestionIndex = 0;}
Quiz.prototype.getQuestionIndex = function() {
        return this.questions[this.QuestionIndex];}
Quiz.prototype.isEnded = function() {
        return this.questions.length === this.QuestionIndex;}
Quiz.prototype.guess = function(answer) {
       if (this.getQuestionIndex().correctAnswer(answer)) {this.score++;}
       this.QuestionIndex++;}
function populate(){
    if (quiz.isEnded()){showScores();}
    else {  //show  question
        var element = document.getElementById("question");
        element.innerHTML=quiz.getQuestionIndex().text;  //show choices
        var choices = quiz.getQuestionIndex().choices;
        for (var i = 0; i < choices.length; i++) {
            var element = document.getElementById("choice" + i);
            element.innerHTML = choices[i];
            guess("btn" + i, choices[i]); }
        showProgress();}};
function guess (id, guess) {
    var button = document.getElementById(id);
    button.onclick = function () {
        quiz.guess(guess);  populate(); }}
function showProgress () {
    var currentQuestionNumber = quiz.QuestionIndex + 1;
    var element = document.getElementById("progress");
    element.innerHTML = "Question " + currentQuestionNumber + " of " +
quiz.questions.length;}
function showScores() {
    var gameOverHtml = "<h1>Result</h1>";
    gameOverHtml += "<h2 id='score' style='margin-top:130px;'> Your scores: " +
quiz.score + " out of " + quiz.questions.length + "</h2>";
    var element = document.getElementById ("quiz");
    element.innerHTML = gameOverHtml;}
       var questions =[
    new Question ("Who is the president of Czech Republic?", ["Donald Trump",
"Andrej Kiska", "Milos Zeman", "Angela Merkel"], "Milos Zeman"),
    new Question ("The capital of Czech Republic is:", ["Brno", "Ostrava", "Hradec
Kralove", "Prague"], "Prague"),
    new Question ("Which food is not common Czech food?", ["Cevapcici", "Svickova",
"Trdelnik", "Tatarak"], "Cevapcici"),
]; // displaying only 3 questions because the code is the same for all of them
var quiz = new Quiz(questions);
populate();(function(){
  window.ScaleRaphael = function(container, width, height){
    var wrapper = document.getElementById(container);
    if (!wrapper.style.position) wrapper.style.position = "relative";
    wrapper.style.width = width + "px";  wrapper.style.height = height + "px";
    wrapper.style.overflow = "hidden";
    var nestedWrapper;
    if (Raphael.type == "VML"){
      wrapper.innerHTML = "<rvml:group style='position : absolute; width: 1000px;
height: 1000px; top: 0px; left: 0px' coordsize='1000,1000' class='rvml'
id='vmlgroup'><\/rvml:group>";
      nestedWrapper = document.getElementById("vmlgroup");
    }else{
      wrapper.innerHTML = "<div id='svggroup'><\/div>";
      nestedWrapper = document.getElementById("svggroup");}
    var paper = new Raphael(nestedWrapper, width, height);
    var vmlDiv;
    if (Raphael.type == "SVG"){
      paper.canvas.setAttribute("viewBox", "0 0 "+width+" "+height);
    }else{
      vmlDiv = wrapper.getElementsByTagName("div")[0];}
    paper.changeSize = function(w, h, center, clipping){
      clipping = !clipping;
      var ratioW = w / width;
      var ratioH = h / height;
      var scale = ratioW < ratioH ? ratioW : ratioH;
      var newHeight = parseInt(height * scale);
      var newWidth = parseInt(width * scale);
      if (Raphael.type == "VML"){// scale the textpaths
       var txt = document.getElementsByTagName("textpath");
        for (var i in txt){
          var curr = txt[i];
          if (curr.style){
            if(!curr._fontSize){
              var mod = curr.style.font.split("px");
              curr._fontSize = parseInt(mod[0]);
              curr._font = mod[1];}
            curr.style.font = curr._fontSize * scale + "px" + curr._font;}}
        var newSize;
        if (newWidth < newHeight){
         newSize = newWidth * 1000 / width;
        }else{
         newSize = newHeight * 1000 / height;}
        newSize = parseInt(newSize);
        nestedWrapper.style.width = newSize + "px";
        nestedWrapper.style.height = newSize + "px";
        if (clipping){
          nestedWrapper.style.left = parseInt((w - newWidth) / 2) + "px";
          nestedWrapper.style.top = parseInt((h - newHeight) / 2) + "px"; }
        vmlDiv.style.overflow = "visible";}
      if (clipping){
        newWidth = w;  newHeight = h; }
      wrapper.style.width = newWidth + "px";
      wrapper.style.height = newHeight + "px";
      paper.setSize(newWidth, newHeight);
      if (center){
        wrapper.style.position = "absolute";
        wrapper.style.left = parseInt((w - newWidth) / 2) + "px";
        wrapper.style.top = parseInt((h - newHeight) / 2) + "px";}}
    paper.scaleAll = function(amount){
```

```
    paper.changeSize(width * amount, height * amount);}
    paper.changeSize(width, height);
    paper.w = width;  paper.h = height;
    return paper; }})();
$(function () {
    var stateNames = new Array();
    var stateURLs = new Array();
    var stateModes = new Array();
    var stateColors = new Array();
    var stateOverColors = new Array();
    var stateClickedColors = new Array();
    var stateText = new Array();
    var offColor;
    var strokeColor;
    var mapWidth;
    var mapHeight;
    var useSideText;
    var textAreaWidth;
    var textAreaPadding;
    var mouseX = 0;
    var mouseY = 0;
    var current = null; // Detect if the browser is IE.
    var IE = $.browser.msie ? true : false;
    $.ajax({
        type: 'GET',
        url: 'xml/settings.xml',
        dataType: $.browser.msie ? 'text' : 'xml',
        success: function (data) {
            var xml;
            if ($.browser.msie) {
                xml = new ActiveXObject('Microsoft.XMLDOM');
                xml.async = false;
                xml.loadXML(data);
            } else {  xml = data;}
            var $xml = $(xml);
            offColor = '#' + $xml.find('mapSettings').attr('offColor');
            strokeColor = '#' + $xml.find('mapSettings').attr('strokeColor');
            mapWidth = $xml.find('mapSettings').attr('mapWidth');
            mapHeight = $xml.find('mapSettings').attr('mapHeight');
            useSideText = $xml.find('mapSettings').attr('useSideText');
            textAreaWidth = $xml.find('mapSettings').attr('textAreaWidth');
            textAreaPadding = $xml.find('mapSettings').attr('textAreaPadding');
            if (useSideText == 'true') {    $("#text").css({
   'width': (parseFloat(textAreaWidth) - parseFloat(textAreaPadding * 2)) + 'px',
   'height': (parseFloat(mapHeight) - parseFloat(textAreaPadding * 2)) + 'px',
   'display': 'inline', 'float': 'right', 'padding':textAreaPadding + 'px' });
                $('#text').html($xml.find('defaultSideText').text());} //Parse xml
            $xml.find('stateData').each(function (i) {
            var $node = $(this);
            stateText.push($node.text());
            stateNames.push($node.attr('stateName'));
            stateURLs.push($node.attr('url'));
            stateModes.push($node.attr('stateMode'));
            stateColors.push('#' + $node.attr('initialStateColor'));
            stateOverColors.push('#' + $node.attr('stateOverColor'));
            stateClickedColors.push('#' + $node.attr('stateSelectedColor')); });
            createMap();} });
    function createMap() {  //start map
        var r = new ScaleRaphael('map', 646, 654),
            attributes = {
                fill: '#d9d9d9',
                cursor: 'pointer',
                stroke: strokeColor,
                'stroke-width': 1,
                'stroke-linejoin': 'round' },
            arr = new Array();
        for (var state in paths) {//Create obj
            var obj = r.path(paths[state].path);
            obj.attr(attributes);
            arr[obj.id] = state;
            if (stateModes[obj.id] == 'OFF') {
                obj.attr({ fill: offColor, cursor: 'default'  });
            } else {   obj.attr({fill: stateColors[obj.id]});
                obj.mouseover(function (e) {
                    if (this != current) {  this.animate({
                            fill: stateOverColors[this.id] }, 500);}
                    var point = this.getBBox(0);
                    $('#map').next('.point').remove();
                    $('#map').after($('<div />').addClass('point'));
                    $('.point').html(stateNames[this.id]).css({
                        left: mouseX - 50,  top: mouseY - 70
                    }).fadeIn();});
                obj.mouseout(function (e) {
                    if (this != current) {
                        this.animate({fill: stateColors[this.id]}, 500); }
                    $('#map').next('.point').remove();});
                obj.mouseup(function (e) {
                    var t = $('#text')[0];
                    t.scrollLeft = 0;
                    t.scrollTop = 0;
                    if (current) {
              current.animate({ fill: stateColors[current.id]   }, 500);  }
              this.animate({ fill: stateClickedColors[this.id]   }, 500);
                    current = this;
           if (useSideText == 'true') {  $('#text').html(stateText[this.id]);
        } else { window.location = stateURLs[this.id];}  });  }}
        resizeMap(r); }
    if (document.captureEvents && Event.MOUSEMOVE) {
        document.captureEvents(Event.MOUSEMOVE); }
    function getMouseXY(e) {  var scrollTop = $(window).scrollTop();
        if (e && e.pageX) {
            mouseX = e.pageX;
            mouseY = e.pageY-scrollTop} else {
            mouseX = event.clientX + document.body.scrollLeft;
            mouseY = event.clientY + document.body.scrollTop; }
        if (mouseX < 0) {mouseX = 0;}
        if (mouseY < 0) { mouseY = 0;}
        $('#map').next('.point').css({
            left: mouseX - 50, top: mouseY - 70  })}
    document.body.onmousemove = getMouseXY;
    function resizeMap(paper) {
        paper.changeSize(mapWidth, mapHeight, true, false);
        if (useSideText == 'true') {
            $(".mapWrapper").css({
      'width': (parseFloat(mapWidth, 10) + parseFloat(textAreaWidth, 10)) + 'px',
                'height': mapHeight + 'px' });
        } else { $(".mapWrapper").css({
         'width': mapWidth + 'px','height': mapHeight + 'px' });  }}});
```